PUNTATORI E VARIABILI ALLOCATE STATICAMENTE

ALGORITMO Puntatori_1

PROCEDURA main ( )

p1, p2, p3 : **PUNTATORE A INT**
a, b, c : **INT**

INIZIO

```
Leggi (a)
Leggi (b)
Leggi (c)

p2 ← &c
p1 ← &b
p3 ← p1

a ← ((*p2) + (*p1)) DIV 3
*p1 ← a * (*p2) – 2*(*p3)
*p2 ← a + (*p1) – (*p2)

p3 ← p2
p2 ← &(*p1)
p1 ← &a

*p1 ← ((*p2) * 4) % (*p3)
*p2 ← (*p1) + (*p2) – 3*(*p3)
*p3 ← ((*p1) – (*p2)) DIV 5

Scrivi (a)
Scrivi (b)
Scrivi (c)

Scrivi (*p1)
Scrivi (*p2)
Scrivi (*p3)

Scrivi (p1)
Scrivi (p2)
Scrivi (p3)
```

RITORNA
FINE

Esempio 1) Dire quale sarà il valore di **a, b, c, *p1, *p2, *p3, p1, p2, p3** dopo avere eseguito lo pseudocodice dell'algoritmo "Puntatori_1" seguente, illustrando il ragionamento eseguito per ottenere il risultato attraverso uno o più disegni esplicativi, nel caso l'utente immetta i seguenti valori iniziali:

1) a = 5, b = 7, c = -2
[R: a = -12 b = 11 c = -4 p1 = &a p2 = &b p3 = &c]

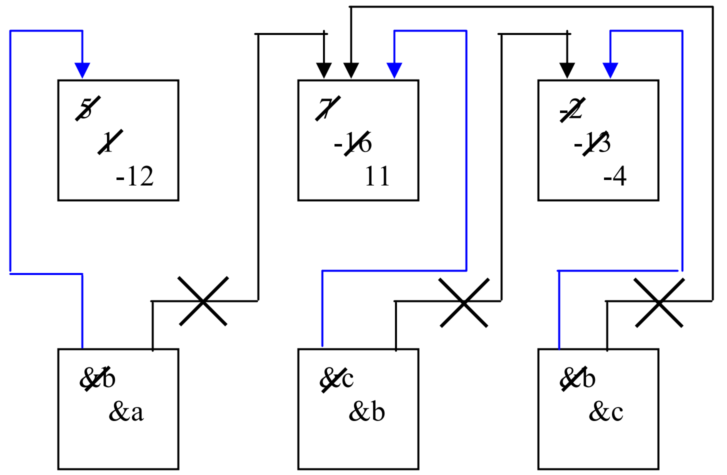
2) a = -5, b = 8, c = 11
[R: a = 20 b = -65 c = 17 p1 = &a p2 = &b p3 = &c]

3) a = 12, b = -6, c = 5
[R: a = 6 b = -3 c = 1 p1 = &a p2 = &b p3 = &c]

4) a = -11, b = 13, c = 14
[R: a = 20 b = -165 c = 37 p1 = &a p2 = &b p3 = &c]

5) a = -7, b = 5, c = 13
[R: a = 28 b = -87 c = 23 p1 = &a p2 = &b p3 = &c]

Segmento Dati



<u>1) Calcoli da eseguire per costruzione dello schema grafico (escluse le istruzioni di I/O)</u>

p2 ← &c (p2 punterà alla variabile c: occorre disegnare la relativa freccia)
p1 ← &b (p1 punterà alla variabile b: occorre disegnare la relativa freccia)
p3 ← p (p3 punterà alla medesima variabile puntata da p1 occorre disegnare la relativa freccia)

a ← ((*p2) + (*p1)) DIV 3 (a = ((-2) + 7) DIV 3 = 5 DIV 3 = 1)
p1 ← a * (*p2) – 2*(*p3) (*p1 = b = 1 * (-2) – 2 * 7 = -2 -14 = - 16)
p3 ← a + (*p1) – (*p2) (*p2 = c = 1 + (-16) – (-2) = 1 – 16 + 2 = -13)

p3 ← p2 (p3 punterà alla medesima variabile puntata da p2: disegnare la nuova freccia e cancellare precedente freccia)
p2 ← &(*p1) (p2 punterà alla medesima variabile puntata da p1 disegnare la nuova freccia e cancellare precedente freccia)
p1 ← &a (p1 punterà alla variabile a: disegnare la nuova freccia e cancellare precedente freccia)

*p1 ← ((*p2) * 4) % (*p3) (*p1 = a = ((-16) * 4) % (-13) = (-64) % (-13) = -12)
*p2 ← (*p1) + (*p2) – 3*(*p3) (*p2 = b = -12 + 8-16) - 3*(-13) = -28 + 39 = 11)
*p3 ← ((*p1) – (*p2)) DIV 5 (*p3 = c = (-12 -11) DIV 5 = (-23) DIV 5 = -4)

Soluzione

a	**b**	**c**	***p1**	***p2**	***p3**	**p1**	**p2**	**p3**
-12	11	-4	-12	11	-4	&a	&b	&c

Esempio 2) Dire quale sarà il valore di **a, b, c, *p1, *p2, *p3, p1, p2, p3** dopo avere eseguito lo pseudocodice dell'algoritmo "Puntatori_2" seguente, illustrando il ragionamento eseguito per ottenere il risultato attraverso uno o più disegni esplicativi, nel caso l'utente immetta i seguenti valori iniziali:

1) a = 5, b = -6, c = 2
[R: a = 56 b = -204 c = -148 p1 = &c p2 = &a p3 = &b]

2) a = 7, b = 5, c = -3
[R: a = 40 b = -66 c = -26 p1 = &c p2 = &a p3 = &b]

3) a = -4, b = -3, c = 8
[R: a = 38 b = 42 c = 80 p1 = &c p2 = &a p3 = &b]

4) a = -11, b = 3, c = -2
[R: a = -12 b = 0 c = -12 p1 = &c p2 = &a p3 = &b]

5) a = 9, b = 1, c = -7
[R: a = 22 b = -26 c = -4 p1 = &c p2 = &a p3 = &b]

ALGORITMO Puntatori_2

PROCEDURA main ( )

p1, p2, p3 : **PUNTATORE A INT**
a, b, c : **INT**

INIZIO

Leggi (a)
Leggi (b)
Leggi (c)

p1 ← &b
p2 ← p1
p3 ← &a

*p1 ← (c * (*p2)) DIV 4
c ← ((*p3) + 2) % ((*p2) – 1)
*p2 ← (c - (*p1)) * ((*p2) – 5)

p3 ← &(*p2)
p2 ← &a
p1 ← &c

a ← (*p1) + (*p2) - (*p3)
*p3 ← (b – c) * ((*p1) + 1)
*p1 ← a + (*p3)

Scrivi (a)
Scrivi (b)
Scrivi (c)

Scrivi (*p1)
Scrivi (*p2)
Scrivi (*p3)

Scrivi (p1)
Scrivi (p2)
Scrivi (p3)

RITORNA
FINE

Esempio 3) Dire quale sarà il valore di **a, b, *p1, *p2, *p, p1, p2, p** dopo avere eseguito lo pseudocodice dell'algoritmo "Puntatori_3" seguente, illustrando il ragionamento eseguito per ottenere il risultato attraverso uno o più disegni esplicativi, nel caso l'utente immetta i seguenti valori iniziali:

1) a = 5, b = -3
[R: a = 3 b = 1 p1 = &b p2 = &b p = &a]

2) a = 3, b = -12
[R: a = 6 b = 2 p1 = &b p2 = &b p = &a]

3) a = -5, b = 4
[R: a = -6 b = -2 p1 = &b p2 = &b p = &a]

4) a = -16, b = 34
[R: a = -18 b = -6 p1 = &b p2 = &b p = &a]

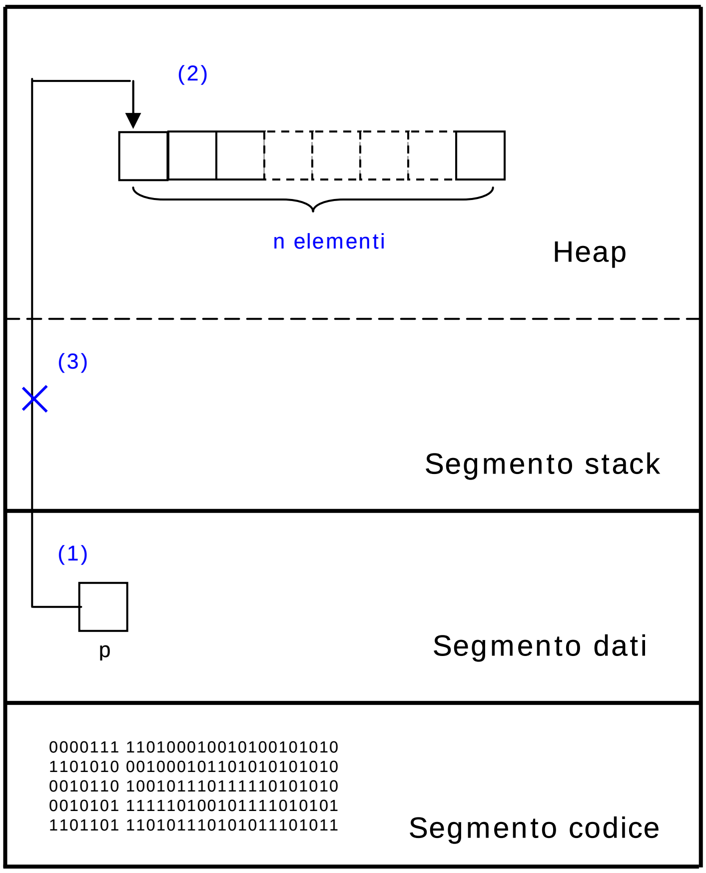
5) a = 4, b = -15
[R: a = 9 b = 3 p1 = &b p2 = &b p = &a]

```
ALGORITMO Puntatori_3
PROCEDURA main ( )
p1, p2, p : PUNTATORE A INT
a, b : INT
INIZIO
Leggi (a)
Leggi (b)
p1 ← &b
p2 ← &a
p ← p1
*p1 ← ((*p2) – (*p1)) % 4
*p2 ← ((*p) + (*p1) – (*p2)) DIV 3
*p ← (*p1) – (*p2)
p2 ← p
p ← &a
*p ← b – (*p1) + 3 * (*p2)
b ← 2 * a – 5 * (*p1)
Scrivi (a)
Scrivi (b)
Scrivi (*p1)
Scrivi (*p2)
Scrivi (*p)
Scrivi (p1)
Scrivi (p2)
Scrivi (p)
RITORNA
FINE
```

PUNTATORI ED ALLOCAZIONE DINAMICA

Esempio 4) Scrivere la pseudocodifica di un algoritmo che esegue il caricamento e la visualizzazione di un vettore o array monodimensionale allocato dinamicamente di n elementi interi

```
ALGORITMO Array_Dinamico_1
PROCEDURA main ( )
p : PUNTATORE A INT
n, i : INT
INIZIO
/* Check sul numero di elementi possibili dell'array */
/* dinamico: VERA DINAMICITA' */
RIPETI
  Leggi (n)
FINCHE' (n ≥ 1 )
/* Allocazione area di memoria dimamica */
Alloca (p , n * DimensioneDi (INT)) (1)
SE (p ≠ NULL)
  ALLORA
    /* Ciclo di caricamento array dinamico */
    PER i ← 0 A (n – 1) ESEGUI
      Leggi (*(p + i)) (2)
      i ← i + 1
     FINE PER
    /* Ciclo di visualizzazione array dinamico */
    PER i ← 0 A (n – 1) ESEGUI
      Scrivi (*(p + i)) (2)
      i ← i + 1
     FINE PER
    /* Deallocazione area di memoria dimamica */
     Dealloca (p) (3)
  ALTRIMENTI
    Scrivi ("Errore nell'allocazione")
RITORNA
FINE
```



(1) La funzione **Alloca** (…), se terminata con esito positivo, collegherà il puntatore **p** ad un'area di memoria allocata nello heap contenente **n** elementi aventi una lunghezza in byte tale da contenere dati del tipo previsto dalla funzione **DimensioneDi** (…).
Nel nostro caso quindi p punterà al primo elemento (di n previsti) in grado di contenere valori interi

N.B. La funzione **Alloca** (…) non inizializza in alcun modo i valori contenuti nelle locazioni di memoria fornite nello heap

(2) Il puntatore **p**, una volta che la funzione **Alloca** (…) ha avuto esito positivo, punterà alla prima locazione di memoria dell'area complessiva assegnata nello heap.
Per poter acccedere agli altri elementi è possibile utilizzare **l'aritmetica dei puntatori**.
In particolare, per quanto riguarda sia il caricamento sia la visualizzazione degli elementi dell'array dinamico, sarà possibile accedere ai vari elementi tenendo presente l'operazione somma di un puntatore ed un intero (in caso di iterazioni con indice crescente) oppure l'operazione differenza di un puntatore ed un intero (in caso di iterazioni con indice decrescente)

In particolare per iterazioni con indice crescente

p ptr alla prima locazione di memoria (ossia **p + 0**) *p (ossia * (p+0)) primo elemento dell'array
p+1 ptr alla seconda locazione di memoria *(p+1) secondo elemento dell'array
p+2 ptr alla terza locazione di memoria *(p+2) terzo elemento dell'array
……..

p+(n-1) ptr alla n-esima locazione di memoria *(p+(n-1)) n-esimo elemento dell'array

(3) La funzione **Dealloca** (…) scollegherà il puntatore **p** dall'area di memoria allocata precedentemente nello heap dalla funzione **Alloca** (…) mettendola a disposizione per eventuali altre allocazioni dinamiche (garbage collection)

N.B. La funzione **Dealloca** (…) non ripulisce in alcun modo i valori precedentemente assegnati

ALGORITMO Array_Dinamico_ERRATO
PROCEDURA main ( )

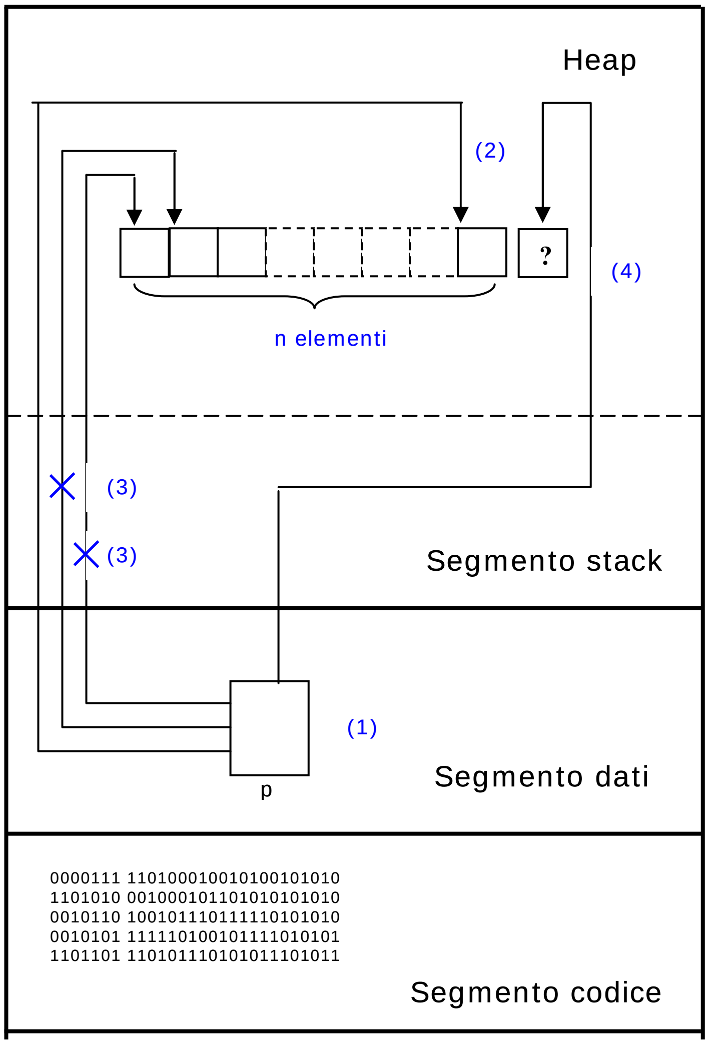
p : **PUNTATORE A INT**
n, i : **INT**

INIZIO

/* Check sul numero di elementi possibili dell'array */
/* dinamico: VERA DINAMICITA' */
RIPETI
  Leggi (n)
FINCHE' ($n \geq 1$)

/* Allocazione area di memoria dinamica */
Alloca (p , n * DimensioneDi (INT)) (1)
SE ($p \neq$ NULL)
  ALLORA
    /* Ciclo di caricamento array dinamico */
    PER i ← 0 **A** (n – 1) **ESEGUI**
      p ← p + i (2)
      Leggi (*p) (3)
      i ← i + 1
    FINE PER
    /* Ciclo di visualizzazione array dinamico */
    PER i ← 0 **A** (n – 1) **ESEGUI**
      p ← p + i (3)
      Scrivi (*p) (4)
      i ← i + 1
    FINE PER
    /* Deallocazione area di memoria dinamica */
    Dealloca (p)
  ALTRIMENTI
    Scrivi ("Errore nell'allocazione")

RITORNA
FINE



```
ALGORITMO Array_Dinamico_2
PROCEDURA main ( )
p, pcur : PUNTATORE A INT
n, i : INT
INIZIO
/* Check sul numero di elementi possibili dell’array */
/* dinamico: VERA DINAMICITA’ */
RIPETI
  Leggi (n)
FINCHE’ (n ≥ 1 )
/* Allocazione area di memoria dimamica */
Alloca (p , n * DimensioneDi (INT))
SE (p ≠ NULL)
  ALLORA
    /* Ciclo di caricamento array dinamico */
    pcur ← p
    PER i ← 0 A (n – 1) ESEGUI
      pcur ← pcur + i
      Leggi (*pcur)
      i ← i + 1
     FINE PER
    /* Ciclo di visualizzazione array dinamico */
    pcur ← p
    PER i ← 0 A (n – 1) ESEGUI
      pcur ← pcur + i
      Scrivi (*pcur)
      i ← i + 1
     FINE PER
    /* Deallocazione area di memoria dimamica */
     Dealloca (p)
  ALTRIMENTI
    Scrivi (“Errore nell’allocazione”)
RITORNA
FINE
```

```
ALGORITMO Array_Dinamico_3
PROCEDURA main ( )
p, pcur : PUNTATORE A INT
n, i : INT
INIZIO
/* Check sul numero di elementi possibili dell'array */
/* dinamico: VERA DINAMICITA' */
RIPETI
  Leggi (n)
FINCHE' (n ≥ 1 )
/* Allocazione area di memoria dimamica */
Alloca (p , n * DimensioneDi (INT))
SE (p ≠ NULL)
  ALLORA
    /* Ciclo di caricamento array dinamico */
    pcur ← p + (n - 1)
    PER i ← (n - 1) INDIETRO A 0 ESEGUI
      Leggi (*pcur)
      pcur ← pcur – (n – i)
      i ← i - 1
    FINE PER
    /* Ciclo di visualizzazione array dinamico */
    pcur ← p
    PER i ← 0 A (n - 1) ESEGUI
      pcur ← pcur + i
      Scrivi (*pcur)
      i ← i + 1
    FINE PER
    /* Deallocazione area di memoria dimamica */
     Dealloca (p)
  ALTRIMENTI
    Scrivi ("Errore nell'allocazione")
RITORNA
FINE
```

Esempio 5) Scrivere la pseudocodifica di un algoritmo che esegue il caricamento e la visualizzazione di un vettore o array monodimensionale allocato staticamente di n elementi interi utilizzando il fatto che il nome dell'array è un puntatore al suo primo elemento.

ALGORITMO Array_Statico_Con_Puntatore_Stile_C

PROCEDURA main ( )

v : ARRAY[MAXDIM] DI INT
n, i : **INT**

INIZIO

/* Check sul numero di elementi possibili dell'array */
/* statico: FALSA DINAMICITA' */
RIPETI
  Leggi (n)
FINCHE' (n ≥ 1) AND (n ≤ MAXDIM)

/* Ciclo di caricamento array statico */
PER i ← 0 **A** (n – 1) **ESEGUI**
  Leggi (*(v + i))
  i ← i + 1
FINE PER

/* Ciclo di visualizzazione array statico */
PER i ← 0 **A** (n - 1) **ESEGUI**
  Scrivi (*(v + i))
  i ← i + 1
FINE PER

RITORNA
FINE

ARITMETICA DEI PUNTATORI

Esempio 6) Sia **p** un puntatore al primo elemento di un array allocato dinamicamente di **20** elementi **interi** e si supponga che **l'indirizzo** contenuto nel puntatore p sia pari al numero decimale **$(1024)_{10}$**.

Supponendo che per il tipo base **intero** vengano utilizzati **2 byte** rispondere alle seguenti domande:

1) Quale è il puntatore al suo undicesimo elemento?

2) A quale indirizzo di memoria espresso in base 10 punterà? Ed in base 16?

3) A quale indirizzo di memoria espresso in base 10 corrisponde il puntatore **p + 7**?

4) A quale indirizzo di memoria espresso in base 16 corrisponde il puntatore **p + 12**?

5) Quale elemento dell'array si trova all'indirizzo di memoria **$(1056)_{10}$** ?

6) Quale elemento dell'array si trova all'indirizzo di memoria **$(426)_{16}$**?

7) Quale elemento dell'array si trova all'indirizzo di memoria **$(1064)_{10}$** ?

8) Quale elemento dell'array si trova all'indirizzo di memoria **$(428)_{16}$**?

9) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = **$(1050)_{10}$** e p2 = **$(1028)_{10}$**?

10) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = **$(420)_{16}$** e p2 = **$(410)_{16}$**?

11) Quale è il puntatore q al suo ultimo elemento?

12) A quale indirizzo di memoria espresso in base 10 punterà? Ed in base 16?

Esempio 7) Sia **q** un puntatore all'ultimo elemento di un array allocato dinamicamente di **20** elementi **interi** e si supponga che **l'indirizzo** contenuto nel puntatore q sia pari al numero decimale **$(1062)_{10}$**.

Supponendo che per il tipo base **intero** vengano utilizzati **2 byte** rispondere alle seguenti domande:

1) Quale è il puntatore al suo undicesimo elemento?

2) A quale indirizzo di memoria espresso in base 10 punterà? Ed in base 16?

3) A quale indirizzo di memoria espresso in base 10 corrisponde il puntatore **q - 7**?

4) A quale indirizzo di memoria espresso in base 16 corrisponde il puntatore **q - 12**?

5) Quale elemento dell'array si trova all'indirizzo di memoria **$(1056)_{10}$** ?

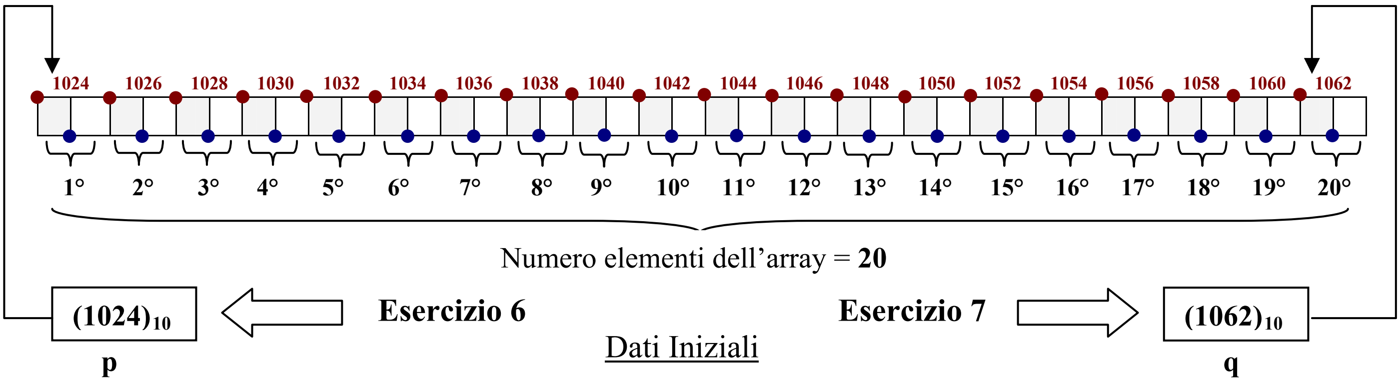
6) Quale elemento dell'array si trova all'indirizzo di memoria **$(426)_{16}$**?

7) Quale elemento dell'array si trova all'indirizzo di memoria **$(1020)_{10}$** ?

8) Quale elemento dell'array si trova all'indirizzo di memoria **$(3FC)_{16}$**?

9) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = **$(1050)_{10}$** e p2 = **$(1028)_{10}$**?

10) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = **$(420)_{16}$** e p2 = **$(410)_{16}$**?



SVOLGIMENTO DELL'ESERCIZIO 6

1) In generale occorre tenere presente che il puntatore **p** assegnato punta al primo elemento dell'array, il puntatore **p + 1** punta al suo secondo elemento, il puntatore **p + 2** punta al suo terzo elemento e che **p + (n - 1)** punta all'ultimo elemento dell'array.

Quindi in generale il puntatore **p + (i - 1)** punta all' **i-esimo elemento dell'array per $1 \le i \le n$** dove **n** è il numero complessivo di elementi dell'array

Quindi il puntatore all'undicesimo elemento dell'array sarà: p + (11 – 1) = **p + 10**

2) L'indirizzo di memoria puntato espresso in base 10 è

p + 10 e per l'aritmetica dei puntatori questa somma corrisponde a
$(1024)_{10}$ + 10 * DimensioneDi (INT) = $(1024)_{10}$ + 10 * 2 = $\mathbf{(1044)_{10}}$

Per rispondere alla seconda parte della domanda basterà effettuare la conversione
$(1044)_{10}$ = (??? $)_{16}$ utilizzando il ben noto **metodo delle divisioni successive** per 16 arrestandosi quando l'ultimo quoziente è nullo

1044: 16
4 65 : 16 il numero convertito si ottiene prendendo i resti dal basso verso l'alto
1 4 : 16
4 **0 STOP**

Quindi $(1044)_{10}$ = $\mathbf{(\ 414\ )_{16}}$

3) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

$0 \le$ spostamento$\le$ n -1 in questo caso SI perchè $0 \le 7 \le 19$

L'indirizzo di memoria puntato dal puntatore **p + 7** per l'aritmetica dei puntatori corrisponde a
$(1024)_{10}$ + 7 * DimensioneDi (INT) = $(1024)_{10}$ + 7 * 2 = $\mathbf{(1038)_{10}}$ (ottavo elemento dell'array)

4) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

$0 \le$ spostamento$\le$ n -1 in questo caso SI perchè $0 \le 12 \le 19$

L'indirizzo di memoria puntato dal puntatore **p + 12** per l'aritmetica dei puntatori corrisponde a
$(1024)_{10}$ + 12 * DimensioneDi (INT) = $(1024)_{10}$ + 12 * 2 = $\mathbf{(1048)_{10}}$

Per rispondere alla domanda basterà effettuare la conversione
$(1048)_{10}$ = (??? $)_{16}$ utilizzando il ben noto metodo delle divisioni successive per 16 arrestandosi quando l'ultimo quoziente è nullo

1048: 16
8 65 : 16 il numero convertito si ottiene prendendo i resti dal basso verso l'alto
1 4 : 16
4 **0 STOP**

Quindi $(1048)_{10}$ = $\mathbf{(\ 418\ )_{16}}$

5) Occorre innanzitutto verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo primo elemento $\mathbf{(1024)_{10}}$** e numero di elementi complessivi dell'array **20** che porta all'**indirizzo dell'ultimo elemento $\mathbf{(1062)_{10}}$** in quanto p + (20 – 1) = p + 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1024)_{10}$ + 19 * DimensioneDi (INT) = $(1024)_{10}$ + 38 = $(1062)_{10}$)

Quindi l'esercizio andrà svolto se l'indirizzo decimale fornito in input è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1024)_{10} \le (x)_{10} \le (1062)_{10}}$$

In questo caso SI perché $\mathbf{(1024)_{10} \le (1056)_{10} \le (1062)_{10}}$

Basta applicare allora la formula

$$\text{Elemento} = \frac{(p_{dato} - p_{iniziale})}{\text{DimensioneDi(Tipo base)}} + 1$$

Nota Bene
Con il risultato **i** tale che **1 ≤ i≤ n**

Nel nostro caso

$$\text{Elemento} = \frac{(1056)_{10} - (1024)_{10}}{2} + 1 = 17$$

All'indirizzo di memoria $(1056)_{10}$ si trova il **17°** elemento dell'array
(N.B. possibile perché l'array ha 20 elementi)

6) Effettueremo la conversione $(426)_{16}$ = $(???)_{10}$ per poter verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo primo elemento $(1024)_{10}$** e numero di elementi complessivi dell'array **20** che porta all'**indirizzo dell'ultimo elemento $(1062)_{10}$** in quanto p + (20 – 1) = p + 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1024)_{10}$ + 19 * DimensioneDi (INT) = $(1024)_{10}$ + 38 = $(1062)_{10}$)

Quindi l'esercizio andrà svolto solo se l'indirizzo esadecimale fornito in input, dopo la conversione in base 10, è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1024)_{10} \le (x)_{10} \le (1062)_{10}}$$

Per effettuare la conversione richiesta si utilizza il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(426)_{16} = 4 * 16^2 + 2 * 16^1 + 6 * 16^0 = 4 * 256 + 2 * 16 + 6 * 1 = 1024 + 32 + 6 = (1062)_{10}$

In questo caso SI perché $(1024)_{10} \le (1062)_{10} \le (1062)_{10}$

Basta applicare allora la formula

$$\text{Elemento} = \frac{(p_{dato} - p_{iniziale})}{\text{DimensioneDi(Tipo base)}} + 1$$

Nel nostro caso

$$\text{Elemento} = \frac{(1062)_{10} - (1024)_{10}}{2} +1 = 20$$

All'indirizzo di memoria $(426)_{16}$ si trova il **20°** elemento dell'array.
(N.B. possibile perché l'array ha 20 elementi)

7) Occorre innanzitutto verificare che l'indirizzo sia coerente con i dati iniziali assegnati (**indirizzo primo elemento $(1024)_{10}$** e numero di elementi complessivi dell'array **20** che porta all'**indirizzo dell'ultimo elemento $(1062)_{10}$** in quanto p + (20 – 1) = p + 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1024)_{10}$ + 19 * DimensioneDi (INT) = $(1024)_{10}$ + 38 = $(1062)_{10}$).

Nel nostro caso l'indirizzo fornito in input **$(1064)_{10}$** non soddisfa il requisito di essere compreso tra **$(1024)_{10}$** indirizzo del primo elemento e **$(1062)_{10}$** indirizzo dell'ultimo elemento dell'array

8) Come nel caso precedente l'esercizio andrà svolto se l'indirizzo esadecimale fornito in input soddisfa il criterio di essere compreso tra **$(1024)_{10}$** indirizzo del primo elemento e **$(1062)_{10}$** indirizzo dell'ultimo elemento dell'array. Controlliamo.

Basta prima effettuare la conversione $(428)_{16}$ = $(???)_{10}$

Per effettuare la conversione richiesta si utilizza il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(428)_{16} = 4 * 16^2 + 2 * 16^1 + 8 * 16^0 = 4 * 256 + 2 * 16 + 8 * 1 = 1024 + 32 + 8 = (1064)_{10}$

Anche in questo caso l'indirizzo fornito in input $\mathbf{(1064)_{10}}$ non soddisfa il requisito di essere compreso tra $\mathbf{(1024)_{10}}$ indirizzo del primo elemento e $\mathbf{(1062)_{10}}$ indirizzo dell'ultimo elemento dell'array

9) Occorre innanzitutto verificare che i due indirizzi siano coerenti con i dati iniziali assegnati (**indirizzo primo elemento** $\mathbf{(1024)_{10}}$ e numero di elementi complessivi dell'array **20** che porta all'**indirizzo dell'ultimo elemento** $\mathbf{(1062)_{10}}$ in quanto p + (20 – 1) = p + 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1024)_{10}$ + 19 * DimensioneDi (INT) = $(1024)_{10}$ + 38 = $(1062)_{10}$)

Quindi i due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1024)_{10} \leq (1028)_{10} \leq (1062)_{10} \quad e \quad (1024)_{10} \leq (1050)_{10} \leq (1062)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

$$\text{Distanza} = \frac{(1050)_{10} - (1028)_{10}}{2} = \frac{22}{2} = \mathbf{11}$$ **ossia ci sono 11 elementi tra i due indirizzi assegnati**

10) Convertiamo dapprima i due numeri esadecimali in base 10

$(420)_{16} = 4 * 16^2 + 2 * 16^1 + 0 * 16^0 = 4 * 256 + 2 * 16 + 0 * 1 = 1024 + 32 = (1056)_{10}$

$(410)_{16} = 4 * 16^2 + 1 * 16^1 + 0 * 16^0 = 4 * 256 + 1 * 16 + 0 * 1 = 1024 + 16 = (1040)_{10}$

Quindi i due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1024)_{10} \leq (1056)_{10} \leq (1062)_{10} \quad e \quad (1024)_{10} \leq (1040)_{10} \leq (1062)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

$$\text{Elemento} = \frac{(1056)_{10} - (1040)_{10}}{2} = \frac{16}{2} = \mathbf{8}$$ **ossia ci sono 11 elementi tra i due indirizzi assegnati**

11) Per quanto detto finora il puntatore all'ultimo elemento dell'array sarà:
q = p + (20 – 1) = **p + 19**

12) L'indirizzo di memoria puntato dal puntatore **q** espresso in base 10 è

p + 19 e per l'aritmetica dei puntatori questa somma corrisponde a
$(1024)_{10}$ + 19 * DimensioneDi (INT) = $(1024)_{10}$ + 19 * 2 = $\mathbf{(1062)_{10}}$

Per rispondere alla seconda parte della domanda basterà effettuare la conversione
$(1062)_{10}$ = (??? $)_{16}$ utilizzando il ben noto **metodo delle divisioni successive** per 16 arrestandosi quando l'ultimo quoziente è nullo

```
1062: 16
   6   66 : 16
        2    4 : 16
             4    0 STOP
```

il numero convertito si ottiene prendendo i resti dal basso verso l'alto

Quindi $(1044)_{10}$ = **(426** $\mathbf{)_{16}}$

SVOLGIMENTO DELL'ESERCIZIO 7

1) In generale occorre tenere presente che il puntatore **q** assegnato punta all'ultimo elemento dell'array, il puntatore **q - 1** punta al penultimo elemento, il puntatore **q - 2** punta al suo terzultimo elemento e che **q - (n - 1)** punta al primo elemento.

Quindi in generale il puntatore **q - (n - i)** punta all' **i-esimo elemento dell'array per $1 \le i \le n$** dove **n** è il numero complessivo di elementi dell'array

Quindi il puntatore all'undicesimo elemento dell'array sarà: q - (20 – 11) = **q - 9**

2) L'indirizzo di memoria puntato espresso in base 10 è

q - 9 e per l'aritmetica dei puntatori questa differenza corrisponde a
$(1062)_{10}$ - 9 * DimensioneDi (INT) = $(1062)_{10}$ - 9 * 2 = $\mathbf{(1044)_{10}}$

Per rispondere alla seconda parte della domanda basterà effettuare la conversione
$(1044)_{10}$ = (??? $)_{16}$ utilizzando il ben noto **metodo delle divisioni successive** per 16 arrestandosi quando l'ultimo quoziente è nullo

```
1044: 16
   4  65 : 16
       1   4 : 16
           4   0  STOP
```

il numero convertito si ottiene prendendo i resti dal basso verso l'alto

Quindi $(1044)_{10}$ = $\mathbf{(414)_{16}}$

3) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

$0 \le$ spostamento$\le$ n -1 in questo caso SI perchè $0 \le 7 \le 19$

L'indirizzo di memoria puntato dal puntatore **q - 7** per l'aritmetica dei puntatori corrisponde a
$(1062)_{10}$ - 7 * DimensioneDi (INT) = $(1062)_{10}$ - 7 * 2 = $\mathbf{(1048)_{10}}$

Per conoscere a quale elemento dell'array corrisponde basterà applicare la seguente formula

Elemento = Numero elementi array - Offset

Nel nostro caso Elemento = n – 7 = 20 – 7 = **13** (tredicesimo elemento dell'array)

4) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

$0 \le$ spostamento$\le$ n -1 in questo caso SI perché $0 \le 12 \le 19$

L'indirizzo di memoria puntato dal puntatore **q - 12** per l'aritmetica dei puntatori corrisponde a
$(1062)_{10}$ - 12 * DimensioneDi (INT) = $(1062)_{10}$ - 12 * 2 = $\mathbf{(1038)_{10}}$

Per conoscere a quale elemento dell'array corrisponde basterà applicare la formula precedente
Elemento = n – 12 = 20 – 12 = **8** (ottavo elemento dell'array)

Completiamo la risposta effettuando la conversione
$(1038)_{10}$ = (??? $)_{16}$ utilizzando il ben noto metodo delle divisioni successive per 16 arrestandosi quando l'ultimo quoziente è nullo

```
1038: 16
  14  64 : 16
       0   4 : 16
           4   0  STOP
```

il numero convertito si ottiene prendendo i resti dal basso verso l'alto

N.B . Ricordiamo che E = 14

Quindi $(1038)_{10}$ = $\mathbf{(40E)_{16}}$

5) Occorre innanzitutto verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo ultimo elemento $(1062)_{10}$** e numero di elementi complessivi dell'array **20** che porta all'**indirizzo del primo elemento $(1024)_{10}$** in quanto q - (20 – 1) = q - 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1062)_{10}$ - 19 * DimensioneDi (INT) = $(1062)_{10}$ - 38 = $(1024)_{10}$)

Quindi l'esercizio andrà svolto se l'indirizzo decimale fornito in input è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1024)_{10} \leq (x)_{10} \leq (1062)_{10}}$$

In questo caso SI perché $(1024)_{10} \leq (1056)_{10} \leq (1062)_{10}$

Basta applicare la formula

$$\text{Elemento} = n - \frac{(p_{finale} - p_{dato})}{\text{DimensioneDi(Tipo base)}}$$

Nota Bene
Con il risultato **i** tale che **$1 \leq i \leq n$**

Nel nostro caso

$$\text{Elemento} = n - \frac{(1062)_{10} - (1056)_{10}}{2} = 20 - 3 = 17$$

All'indirizzo di memoria $(1056)_{10}$ si trova il **17°** elemento dell'array
(N.B. possibile perché l'array ha 20 elementi)

6) Effettueremo la conversione $(426)_{16} = (???)_{10}$ per poter verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo ultimo elemento $(1062)_{10}$** e numero di elementi complessivi dell'array **20** che porta all'**indirizzo del primo elemento $(1024)_{10}$** in quanto q - (20 – 1) = q - 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1062)_{10}$ - 19 * DimensioneDi (INT) = $(1062)_{10}$ - 38 = $(1024)_{10}$)

Quindi l'esercizio andrà svolto solo se l'indirizzo esadecimale fornito in input, dopo la conversione in base 10, è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1024)_{10} \leq (x)_{10} \leq (1062)_{10}}$$

Per effettuare la conversione richiesta si utilizza il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(426)_{16} = 4 * 16^2 + 2 * 16^1 + 6 * 16^0 = 4 * 256 + 2 * 16 + 6 * 1 = 1024 + 32 + 6 = (1062)_{10}$

In questo caso SI perché $(1024)_{10} \leq (1062)_{10} \leq (1062)_{10}$

Basta applicare allora la formula

$$\text{Elemento} = n - \frac{(p_{finale} - p_{dato})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

$$\text{Elemento} = n - \frac{(1062)_{10} - (1062)_{10}}{2} = 20 - 0 = 20$$

Nota Bene
Con il risultato **i** tale che **$1 \leq i \leq n$**

All'indirizzo di memoria $(426)_{16}$ si trova il **20°** elemento dell'array.
(N.B. possibile perché l'array ha 20 elementi)

7) Occorre innanzitutto verificare che l'indirizzo sia coerente con i dati iniziali assegnati (**indirizzo ultimo elemento $(1062)_{10}$** e numero di elementi complessivi dell'array **20** che porta all'**indirizzo del primo elemento $(1024)_{10}$** in quanto q - (20 – 1) = q - 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1062)_{10}$ - 19 * DimensioneDi (INT) = $(1062)_{10}$ - 38 = $(1024)_{10}$)

Nel nostro caso l'indirizzo fornito in input $\mathbf{(1020)_{10}}$ non soddisfa il requisito di essere compreso tra $\mathbf{(1024)_{10}}$ indirizzo del primo elemento e $\mathbf{(1062)_{10}}$ indirizzo dell'ultimo elemento dell'array

8) Come nel caso precedente l'esercizio andrà svolto se l'indirizzo esadecimale fornito in input soddisfa il criterio di essere compreso tra $\mathbf{(1062)_{10}}$ indirizzo dell'ultimo elemento e $\mathbf{(1024)_{10}}$ indirizzo del primo elemento dell'array. Controlliamo.

Basta prima effettuare la conversione $(3FC)_{16} = (???)_{10}$ utilizzando il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(3FC)_{16} = 3 * 16^2 + F * 16^1 + C * 16^0 = 3 * 16^2 + 15 * 16^1 + 12 * 16^0 = 3 * 256 + 15 * 16 + 12 * 1 = 768 + 240 + 12 = (1020)_{10}$

Anche in questo caso l'indirizzo fornito in input $\mathbf{(1020)_{10}}$ non soddisfa il requisito di essere compreso tra $\mathbf{(1024)_{10}}$ indirizzo del primo elemento e $\mathbf{(1062)_{10}}$ indirizzo dell'ultimo elemento dell'array

9) Occorre innanzitutto verificare che i due indirizzi siano coerenti con i dati iniziali assegnati (**indirizzo ultimo elemento** $\mathbf{(1062)_{10}}$ e numero di elementi complessivi dell'array **20** che porta all'**indirizzo del primo elemento** $\mathbf{(1024)_{10}}$ in quanto q - (20 – 1) = q - 19 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1062)_{10}$ - 19 * DimensioneDi (INT) = $(1062)_{10}$ - 38 = $(1024)_{10}$)

Quindi i due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1024)_{10} \le (1028)_{10} \le (1062)_{10} \quad e \quad (1024)_{10} \le (1050)_{10} \le (1062)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

$$\text{Distanza} = \frac{(1050)_{10} - (1028)_{10}}{2} = \frac{22}{2} = \mathbf{11}$$ **ossia ci sono 11 elementi tra i due indirizzi assegnati**

10) Convertiamo dapprima i due numeri esadecimali in base 10

$(420)_{16} = 4 * 16^2 + 2 * 16^1 + 0 * 16^0 = 4 * 256 + 2 * 16 + 0 * 1 = 1024 + 32 = (1056)_{10}$

$(410)_{16} = 4 * 16^2 + 1 * 16^1 + 0 * 16^0 = 4 * 256 + 1 * 16 + 0 * 1 = 1024 + 16 = (1040)_{10}$

Quindi i due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1024)_{10} \le (1056)_{10} \le (1062)_{10} \quad e \quad (1024)_{10} \le (1040)_{10} \le (1062)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

$$\text{Elemento} = \frac{(1056)_{10} - (1040)_{10}}{2} = \frac{16}{2} = \mathbf{8}$$ **ossia ci sono 8 elementi tra i due indirizzi assegnati**

Esempio 8) Sia **p** un puntatore al primo elemento di un array allocato dinamicamente di **15** elementi **interi** e si supponga che **l'indirizzo** contenuto nel puntatore p sia pari al numero decimale $\mathbf{(1456)_{10}}$.

Supponendo che per il tipo base **intero** vengano utilizzati **4 byte** rispondere alle seguenti domande:

1) Quale è il puntatore al suo ottavo elemento?

2) A quale indirizzo di memoria espresso in base 10 punterà? Ed in base 16?

3) A quale indirizzo di memoria espresso in base 10 corrisponde il puntatore **p + 11**?

4) A quale indirizzo di memoria espresso in base 16 corrisponde il puntatore **p + 3**?

5) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(1488)_{10}}$?

6) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(5D8)_{16}}$?

7) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(1520)_{10}}$? ed all'indirizzo $\mathbf{(1452)_{16}}$?

8) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(5F4)_{16}}$? ed all'indirizzo $\mathbf{(5AC)_{16}}$?

9) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = $\mathbf{(1504)_{10}}$ e p2 = $\mathbf{(1472)_{10}}$?

10) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = $\mathbf{(5E0)_{16}}$ e p2 = $\mathbf{(5C0)_{16}}$?

11) Quale è il puntatore q al suo ultimo elemento?

12) A quale indirizzo di memoria espresso in base 10 punterà? Ed in base 16?

Esempio 9) Sia **q** un puntatore all'ultimo elemento di un array allocato dinamicamente di **15** elementi **interi** e si supponga che **l'indirizzo** contenuto nel puntatore q sia pari al numero decimale $\mathbf{(1512)_{10}}$.

Supponendo che per il tipo base **intero** vengano utilizzati **4 byte** rispondere alle seguenti domande:

1) Quale è il puntatore al suo quarto elemento?

2) A quale indirizzo di memoria espresso in base 10 punterà? Ed in base 16?

3) A quale indirizzo di memoria espresso in base 10 corrisponde il puntatore **q - 5**?

4) A quale indirizzo di memoria espresso in base 16 corrisponde il puntatore **q - 14**?

5) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(1488)_{10}}$?

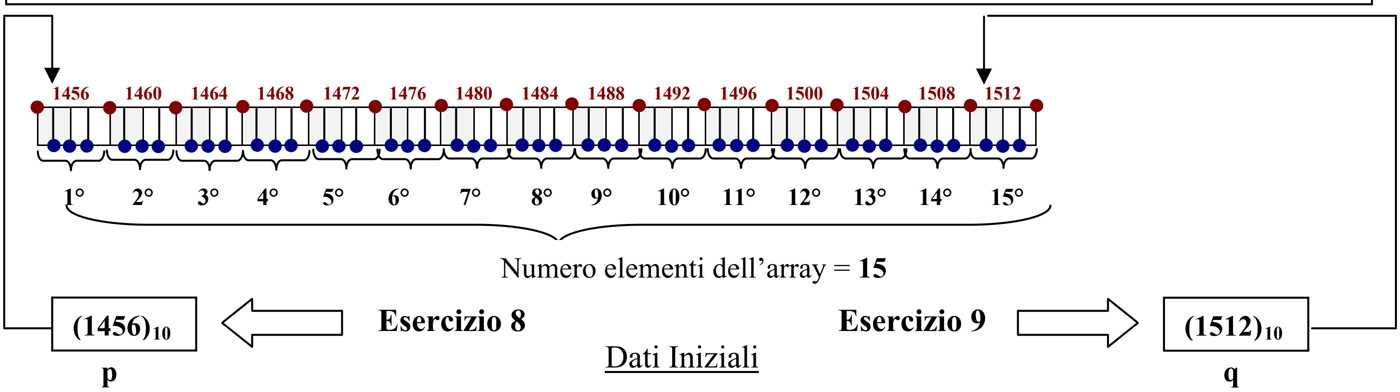
6) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(5E0)_{16}}$?

7) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(1524)_{10}}$? ed all'indirizzo $\mathbf{(1448)_{10}}$?

8) Quale elemento dell'array si trova all'indirizzo di memoria $\mathbf{(5F4)_{16}}$? ed all'indirizzo $\mathbf{(5A8)_{16}}$?

9) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = $\mathbf{(1492)_{10}}$ e p2 = $\mathbf{(1472)_{10}}$?

10) Quanti elementi dell'array si trovano tra i due indirizzi di memoria p1 = $\mathbf{(5BC)_{16}}$ e p2 = $\mathbf{(5DC)_{16}}$?



SVOLGIMENTO DELL'ESERCIZIO 8

1) In generale occorre tenere presente che il puntatore **p** assegnato punta al primo elemento dell'array, il puntatore **p + 1** punta al suo secondo elemento, il puntatore **p + 2** punta al suo terzo elemento e che **p + (n - 1)** punta all'ultimo elemento dell'array.

Quindi in generale il puntatore **p + (i - 1)** punta all' **i-esimo elemento dell'array per 1 ≤ i ≤ n** dove **n** è il numero complessivo di elementi dell'array

Quindi il puntatore all'undicesimo elemento dell'array sarà: p + (8 – 1) = **p + 7**

2) L'indirizzo di memoria puntato espresso in base 10 è

p + 7 e per l'aritmetica dei puntatori questa somma corrisponde a
$(1456)_{10}$ + 7 * DimensioneDi (INT) = $(1456)_{10}$ + 7 * 4 = $\mathbf{(1484)_{10}}$

Per rispondere alla seconda parte della domanda basterà effettuare la conversione
$(1044)_{10}$ = (??? $)_{16}$ utilizzando il ben noto **metodo delle divisioni successive** per 16 arrestandosi quando l'ultimo quoziente è nullo

1484: 16
C 92 : 16 — il numero convertito si ottiene prendendo i resti dal basso verso l'alto
C 5 : 16
5 **0 STOP**

Quindi $(1484)_{10}$ = **(5CC)**$_{\mathbf{16}}$

3) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

0 ≤ spostamento≤ n -1 in questo caso SI perchè 0 ≤ 11≤ 14

L'indirizzo di memoria puntato dal puntatore **p + 11** per l'aritmetica dei puntatori corrisponde a $(1456)_{10}$ + 11 * DimensioneDi (INT) = $(1456)_{10}$ + 11 * 4 = $\mathbf{(1500)_{10}}$ (dodicesimo elemento dell'array)

4) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

0 ≤ spostamento≤ n -1 in questo caso SI perchè 0 ≤ 3≤ 14

$(1456)_{10}$ + 3 * DimensioneDi (INT) = $(1456)_{10}$ + 3 * 4 = $\mathbf{(1468)_{10}}$ (quarto elemento dell'array)

Per rispondere alla domanda basterà effettuare la conversione
$(1468)_{10}$ = (??? $)_{16}$ utilizzando il ben noto metodo delle divisioni successive per 16 arrestandosi quando l'ultimo quoziente è nullo

1468: 16
C 91 : 16 — il numero convertito si ottiene prendendo i resti dal basso verso l'alto
B 5 : 16
5 **0 STOP**

Quindi $(1468)_{10}$ = **(5BC)**$_{\mathbf{16}}$

5) Occorre innanzitutto verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo primo elemento** $\mathbf{(1456)_{10}}$ e numero di elementi complessivi dell'array **15** che porta all'**indirizzo dell'ultimo elemento** $\mathbf{(1512)_{10}}$ in quanto p + (15 – 1) = p + 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1456)_{10}$ + 14 * DimensioneDi (INT) = $(1456)_{10}$ + 56 = $(1512)_{10}$)

Quindi l'esercizio andrà svolto se l'indirizzo decimale fornito in input è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1456)_{10} \le (x)_{10} \le (1512)_{10}}$$

In questo caso SI perché $(1456)_{10} \leq (1488)_{10} \leq (1512)_{10}$

Basta applicare allora la formula

$$\text{Elemento} = \frac{(p_{dato} - p_{iniziale})}{\text{DimensioneDi(Tipo base)}} + 1$$

Nel nostro caso

$$\text{Elemento} = \frac{(1488)_{10} - (1456)_{10}}{2} + 1 = 8 + 1 = 9$$

Nota Bene
Con il risultato **i** tale che **$1 \leq i \leq n$**

All'indirizzo di memoria $(1488)_{10}$ si trova il **9°** elemento dell'array
(N.B. possibile perché l'array ha 15 elementi)

6) Effettueremo la conversione $(5D8)_{16} = (???)_{10}$ per poter verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo primo elemento $(1456)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo dell'ultimo elemento $(1512)_{10}$** in quanto p + (15 – 1) = p + 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1456)_{10}$ + 14 * DimensioneDi (INT) = $(1456)_{10}$ + 56 = $(1512)_{10}$)

Quindi l'esercizio andrà svolto solo se l'indirizzo esadecimale fornito in input, dopo la conversione in base 10, è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1456)_{10} \leq (x)_{10} \leq (1512)_{10}}$$

Per effettuare la conversione richiesta si utilizza il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(5D8)_{16} = 5 * 16^2 + D * 16^1 + 8 * 16^0 = 5 * 256 + 13 * 16 + 8 * 1 = 1280 + 208 + 8 = (1496)_{10}$

In questo caso SI perché $(1456)_{10} \leq (1496)_{10} \leq (1512)_{10}$

Basta applicare allora la formula

$$\text{Elemento} = \frac{(p_{dato} - p_{iniziale})}{\text{DimensioneDi(Tipo base)}} + 1$$

Nel nostro caso

$$\text{Elemento} = \frac{(1496)_{10} - (1456)_{10}}{2} + 1 = 10 + 1 = 11$$

All'indirizzo di memoria $(5D8)_{16}$ si trova l'**11°** elemento dell'array.
(N.B. possibile perché l'array ha 15 elementi)

7) Occorre innanzitutto verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo primo elemento $(1456)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo dell'ultimo elemento $(1512)_{10}$** in quanto p + (15 – 1) = p + 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1456)_{10}$ + 14 * DimensioneDi (INT) = $(1456)_{10}$ + 56 = $(1512)_{10}$)

Nel nostro caso entrambi gli indirizzi **$(1452)_{10}$** e **$(1520)_{10}$** non soddisfano il requisito di essere compresi tra **$(1456)_{10}$** indirizzo del primo elemento e **$(1512)_{10}$** indirizzo dell'ultimo elemento dell'array

8) Occorre innanzitutto verificare che i due indirizzi, dopo la conversione, siano coerenti con i dati iniziali assegnati (**indirizzo primo elemento $(1456)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo dell'ultimo elemento $(1512)_{10}$** in quanto p + (15 – 1) = p + 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1456)_{10}$ + 14 * DimensioneDi (INT) = $(1456)_{10}$ + 56 = $(1512)_{10}$). Controlliamo.

Partiamo dal primo indirizzo esadecimale fornito in input.
Effettuiamo la conversione $(5F4)_{16} = (???)_{10}$

Per effettuare la conversione richiesta si utilizza il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(5F4)_{16} = 5 * 16^2 + F * 16^1 + 4 * 16^0 = 5 * 256 + 15 * 16 + 4 * 1 = 1280 + 240 + 4 = (1524)_{10}$

In questo caso l'indirizzo fornito in input $\mathbf{(1524)_{10}}$ non soddisfa il requisito di essere compreso tra $\mathbf{(1456)_{10}}$ indirizzo del primo elemento e $\mathbf{(1512)_{10}}$ indirizzo dell'ultimo elemento dell'array (sfondamento superiore dell'array).
Continuiamo con il secondo indirizzo esadecimale fornito in input.
Effettuiamo la conversione $(5AC)_{16} = (???)_{10}$

Per effettuare la conversione richiesta si utilizza il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(5AC)_{16} = 5 * 16^2 + A * 16^1 + C * 16^0 = 5 * 256 + 10 * 16 + 12 * 1 = 1280 + 160 + 12 = (1452)_{10}$

In questo caso l'indirizzo fornito in input $\mathbf{(1452)_{10}}$ non soddisfa il requisito di essere compreso tra $\mathbf{(1456)_{10}}$ indirizzo del primo elemento e $\mathbf{(1512)_{10}}$ indirizzo dell'ultimo elemento dell'array (sfondamento inferiore dell'array).

9) Occorre innanzitutto verificare che i due indirizzi siano coerenti con i dati iniziali assegnati (**indirizzo primo elemento** $\mathbf{(1456)_{10}}$ e numero di elementi complessivi dell'array **15** che porta all'**indirizzo dell'ultimo elemento** $\mathbf{(1512)_{10}}$ in quanto p + (15 – 1) = p + 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1456)_{10}$ + 14 * DimensioneDi (INT) = $(1456)_{10}$ + 56 = $(1512)_{10}$)

Quindi i due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1456)_{10} \leq (1472)_{10} \leq (1512)_{10} \quad e \quad (1456)_{10} \leq (1504)_{10} \leq (1512)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

$$\text{Distanza} = \frac{(1504)_{10} - (1472)_{10}}{4} = \frac{32}{4} = \mathbf{8}$$ **ossia ci sono 8 elementi tra i due indirizzi assegnati**

10) Convertiamo dapprima i due numeri esadecimali in base 10

$(5E0)_{16} = 5 * 16^2 + E * 16^1 + 0 * 16^0 = 5 * 256 + 14 * 16 + 0 * 1 = 1280 + 224 = (1504)_{10}$

$(5C0)_{16} = 5 * 16^2 + C * 16^1 + 0 * 16^0 = 5 * 256 + 13 * 16 + 0 * 1 = 1280 + 208 = (1488)_{10}$

Quindi i due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1456)_{10} \leq (1504)_{10} \leq (1512)_{10} \quad e \quad (1456)_{10} \leq (1488)_{10} \leq (1512)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

$$\text{Elemento} = \frac{(1504)_{10} - (1488)_{10}}{4} = \frac{16}{4} = \mathbf{4}$$ **ossia ci sono 4 elementi tra i due indirizzi assegnati**

11) Per quanto detto finora il puntatore all'ultimo elemento dell'array sarà:
q = p + (15 – 1) = **p + 14**

12) L'indirizzo di memoria puntato dal puntatore **q** espresso in base 10 è

p + 14 e per l'aritmetica dei puntatori questa somma corrisponde a
$(1456)_{10}$ + 14 * DimensioneDi (INT) = $(1456)_{10}$ + 14 * 4 = $\mathbf{(1512)_{10}}$

Per rispondere alla seconda parte della domanda basterà effettuare la conversione
$(1512)_{10}$ = (??? $)_{16}$ utilizzando il ben noto **metodo delle divisioni successive** per 16 arrestandosi quando l'ultimo quoziente è nullo

1512: 16
8 94 : 16 il numero convertito si ottiene prendendo i resti dal basso verso l'alto
E 5 : 16 **N.B . Ricordiamo che E = 14**
5 **0 STOP**

Quindi $(1512)_{10}$ = $\mathbf{(\ 5E8\ )_{16}}$

SVOLGIMENTO DELL'ESERCIZIO 9

1) In generale occorre tenere presente che il puntatore **q** assegnato punta all'ultimo elemento dell'array, il puntatore **q - 1** punta al penultimo elemento, il puntatore **q - 2** punta al suo terzultimo elemento e che **q - (n - 1)** punta al primo elemento.

Quindi in generale il puntatore **q - (n - i)** punta all' **i-esimo elemento dell'array per 1 ≤ i ≤ n** dove **n** è il numero complessivo di elementi dell'array

Quindi il puntatore al quarto elemento dell'array sarà: q - (15 – 4) = **q - 11**

2) L'indirizzo di memoria puntato espresso in base 10 è

q - 11 e per l'aritmetica dei puntatori questa differenza corrisponde a
$(1512)_{10}$ - 11 * DimensioneDi (INT) = $(1512)_{10}$ - 11 * 4 = $\mathbf{(1468)_{10}}$

Per rispondere alla seconda parte della domanda basterà effettuare la conversione
$(1468)_{10}$ = (??? $)_{16}$ utilizzando il ben noto **metodo delle divisioni successive** per 16 arrestandosi quando l'ultimo quoziente è nullo

1468: 16
C 91 : 16 il numero convertito si ottiene prendendo i resti dal basso verso l'alto
B 5 : 16
5 **0 STOP**

Quindi $(1468)_{10}$ = $\mathbf{(\ 5BC\ )_{16}}$

3) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

0 ≤ spostamento≤ n -1 in questo caso SI perchè 0 ≤ 5 ≤ 14

L'indirizzo di memoria puntato dal puntatore **q - 5** per l'aritmetica dei puntatori corrisponde a
$(1512)_{10}$ - 5 * DimensioneDi (INT) = $(1512)_{10}$ - 5 * 4 = $\mathbf{(1492)_{10}}$

Per conoscere a quale elemento dell'array corrisponde basterà applicare la seguente formula

Elemento = Numero elementi array - Offset

Nel nostro caso Elemento = n – 5 = 15 – 5 = **10** (decimo elemento dell'array)

4) N.B. Ovviamente l'esercizio andrà svolto se lo spostamento (offset) fornito in input al puntatore è tale che

0 ≤ spostamento≤ n -1 in questo caso SI perché 0 ≤ 14 ≤ 14

L'indirizzo di memoria puntato dal puntatore **q - 14** per l'aritmetica dei puntatori corrisponde a
$(1512)_{10}$ - 14 * DimensioneDi (INT) = $(1512)_{10}$ - 14 * 4 = $\mathbf{(1456)_{10}}$

Per conoscere a quale elemento dell'array corrisponde basterà applicare la formula precedente
Elemento = n – 14 = 15 – 14 = **1** (primo elemento dell'array)

Completiamo la risposta effettuando la conversione
$(1456)_{10}$ = (??? $)_{16}$ utilizzando il ben noto metodo delle divisioni successive per 16 arrestandosi quando l'ultimo quoziente è nullo

1456: 16
0 91 : 16 il numero convertito si ottiene prendendo i resti dal basso verso l'alto
B 5 : 16
5 **0 STOP** **N.B . Ricordiamo che B = 11**

Quindi $(1456)_{10}$ = **(5B0 $)_{16}$**

5) Occorre innanzitutto verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo ultimo elemento $(1512)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo del primo elemento $(1456)_{10}$** in quanto q - (15 – 1) = q - 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1512)_{10}$ - 14 * DimensioneDi (INT) = $(1512)_{10}$ - 56 = $(1456)_{10}$)

Quindi l'esercizio andrà svolto se l'indirizzo decimale fornito in input è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1456)_{10} \le (x)_{10} \le (1512)_{10}}$$

In questo caso SI perché $(1456)_{10} \le (1488)_{10} \le (1512)_{10}$

Basta applicare la formula

$$\text{Elemento} = n - \frac{(p_{finale} - p_{dato})}{\text{DimensioneDi(Tipo base)}}$$

Nota Bene Con il risultato **i** tale che **1 ≤ i≤ n**

Nel nostro caso

$$\text{Elemento} = n - \frac{(1512)_{10} - (1488)_{10}}{4} = 15 - \frac{24}{4} = 15 - 6 = 9$$

All'indirizzo di memoria $(1488)_{10}$ si trova il **9°** elemento dell'array
<u>(N.B. possibile perché l'array ha 15 elementi)</u>

6) Effettueremo la conversione $(5E0)_{16}$ = $(???)_{10}$ per poter verificare che l'indirizzo assegnato sia coerente nello scenario prospettato dall'esercizio (**indirizzo ultimo elemento $(1512)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo del primo elemento $(1456)_{10}$** in quanto q - (15 – 1) = q - 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1512)_{10}$ - 14 * DimensioneDi (INT) = $(1512)_{10}$ - 56 = $(1456)_{10}$)

Quindi l'esercizio andrà svolto solo se l'indirizzo esadecimale fornito in input, dopo la conversione in base 10, è maggiore o uguale rispetto all'indirizzo del primo e minore uguale dell'indirizzo dell'ultimo elemento ossia

$$\mathbf{(1456)_{10} \le (x)_{10} \le (1512)_{10}}$$

Per effettuare la conversione richiesta si utilizza il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16
$(5E0)_{16} = 5 * 16^2 + E * 16^1 + 0 * 16^0 = 5 * 256 + 14 * 16 + 0 * 1 = 1280 + 224 + 0 = (1504)_{10}$

In questo caso SI perché $(1456)_{10} \le (1504)_{10} \le (1512)_{10}$

Basta applicare allora la formula

$$\text{Elemento} = n - \frac{(p_{finale} - p_{dato})}{\text{DimensioneDi(Tipo base)}}$$

Nota Bene Con il risultato **i** tale che **1 ≤ i ≤ n**

Nel nostro caso

$$\text{Elemento} = n - \frac{(1512)_{10} - (1504)_{10}}{4} = 15 - \frac{8}{4} = 15 - 2 = 13$$

All'indirizzo di memoria $(5E0)_{16}$ si trova il **13°** elemento dell'array.
(N.B. possibile perché l'array ha 15 elementi)

7) Occorre innanzitutto verificare che gli indirizzi assegnati siano coerenti con i dati iniziali assegnati (**indirizzo ultimo elemento $(1512)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo del primo elemento $(1456)_{10}$** in quanto q - (15 – 1) = q - 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1512)_{10}$ - 14 * DimensioneDi (INT) = $(1512)_{10}$ - 56 = $(1456)_{10}$). Controlliamo.

Nel nostro caso sia l'indirizzo **$(1524)_{10}$** (sfondamento superiore dell'array), sia l'indirizzo **$(1448)_{10}$** (sfondamento inferiore dell'array) forniti in input non soddisfano il requisito di essere compreso tra **$(1456)_{10}$** indirizzo del primo elemento e **$(1512)_{10}$** indirizzo dell'ultimo elemento dell'array

8) Occorre innanzitutto verificare che gli indirizzi assegnati, dopo la conversione, siano coerenti con i dati iniziali assegnati (**indirizzo ultimo elemento $(1512)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo del primo elemento $(1456)_{10}$** in quanto q - (15 – 1) = q - 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1512)_{10}$ - 14 * DimensioneDi (INT) = $(1512)_{10}$ - 56 = $(1456)_{10}$). Controlliamo.

Basta prima effettuare la conversione $(5F4)_{16} = (???)_{10}$ utilizzando il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(5F4)_{16} = 5 * 16^2 + F * 16^1 + 4 * 16^0 = 5 * 16^2 + 15 * 16^1 + 4 * 16^0 = 5 * 256 + 15 * 16 + 4 * 1 =$
$1280 + 240 + 4 = (1524)_{10}$

In questo primo caso l'indirizzo fornito in input **$(1524)_{10}$** non soddisfa il requisito di essere compreso tra **$(1456)_{10}$** indirizzo del primo elemento e **$(1512)_{10}$** indirizzo dell'ultimo elemento dell'array (sfondamento superiore dell'array).

Ora effettuiamo la conversione $(5A8)_{16} = (???)_{10}$ utilizzando il ben noto **metodo della notazione espansa** che consiste nel calcolare per poi sommare il valore pesato di ogni cifra nella base 16

$(5A8)_{16} = 5 * 16^2 + A * 16^1 + 8 * 16^0 = 5 * 16^2 + 10 * 16^1 + 8 * 16^0 = 5 * 256 + 10 * 16 + 8 * 1 =$
$1280 + 160 + 8 = (1448)_{10}$

Anche in questo secondo caso l'indirizzo fornito in input **$(1448)_{10}$** non soddisfa il requisito di essere compreso tra **$(1456)_{10}$** indirizzo del primo elemento e **$(1512)_{10}$** indirizzo dell'ultimo elemento dell'array (sfondamento inferiore dell'array).

9) Occorre innanzitutto verificare che i due indirizzi siano coerenti con i dati iniziali assegnati (**indirizzo ultimo elemento $(1512)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo del primo elemento $(1456)_{10}$** in quanto q - (15 – 1) = q - 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1512)_{10}$ - 14 * DimensioneDi (INT) = $(1512)_{10}$ - 56 = $(1456)_{10}$)

Nel nostro caso i due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1456)_{10} \leq (1492)_{10} \leq (1512)_{10} \quad e \quad (1456)_{10} \leq (1472)_{10} \leq (1512)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

Distanza $= \frac{(1492)_{10} - (1472)_{10}}{4} = \frac{20}{2}$ = **5** **ossia ci sono 5 elementi tra i due indirizzi assegnati**

10) Occorre innanzitutto verificare che i due indirizzi, dopo la conversione, siano coerenti con i dati iniziali assegnati (**indirizzo ultimo elemento $(1512)_{10}$** e numero di elementi complessivi dell'array **15** che porta all'**indirizzo del primo elemento $(1456)_{10}$** in quanto q - (15 – 1) = q - 14 che porta secondo l'aritmetica dei puntatori all'indirizzo $(1512)_{10}$ - 14 * DimensioneDi (INT) = $(1512)_{10}$ - 56 = $(1456)_{10}$).

Convertiamo dapprima i due numeri esadecimali in base 10

$(5BC)_{16} = 5 * 16^2 + B * 16^1 + C * 16^0 = 5 * 256 + 11 * 16 + 12 * 1 = 1280 + 176 + 12 = (1468)_{10}$

$(5DC)_{16} = 5 * 16^2 + D * 16^1 + C * 16^0 = 5 * 256 + 13 * 16 + 12 * 1 = 1280 + 208 + 12 = (1500)_{10}$

I due valori assegnati sono coerenti con i dati iniziali in quanto entrambi compresi tra l'indirizzo del primo elemento e dell'ultimo elemento dell'array.

$$\mathbf{(1456)_{10} \le (1468)_{10} \le (1512)_{10} \quad e \quad (1456)_{10} \le (1500)_{10} \le (1512)_{10}}$$

Per ottenere il numero di elementi richiesto occorre applicare la formula

$$\text{Distanza} = \frac{(p_{maggiore} - p_{minore})}{\text{DimensioneDi(Tipo base)}}$$

Nel nostro caso

Elemento $= \frac{(1500)_{10} - (1468)_{10}}{4} = \frac{32}{4}$ = **8** **ossia ci sono 8 elementi tra i due indirizzi assegnati**