

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO A**

Si scriva un programma C che

- legga da input due sequenze di 4 numeri interi minori di 10 (si controlli che siano minori di 10). I numeri letti vanno memorizzati in due vettori A e B.
- Crei un vettore SOMMA che contiene in ogni elemento SOMMA[i]

$$SOMMA[i] = \sum_{j=1}^{A[i]} B[i] * j$$

ogni elemento di somma si calcola con  
`int sommatoria(int a, int b);`

che restituisce

$$\sum_{j=1}^a b*j$$

- Stampi a video il contenuto di SOMMA

ESEMPIO:  
 Numeri inseriti  
 2  
 3  
 1  
 4  
 3  
 4  
 2  
 1

I vettori A e B saranno:

A	2	3	1	4
B	3	4	2	1

SOMMA	9	24	2	10
-------	---	----	---	----

**COMPITO A**

```
#include <stdio.h>
#define N 4

int sommatoria(int a, int b)
{int j, sommatoria=0;
  for (j=1;j<=a;j++)
    sommatoria+=b*j;
  return sommatoria;
}

void main()
{int A[N],B[N],SOMMA[N],i=0;
  while (i<N)
    {printf("Elemento n.%d del primo vettore : ",i);
     scanf("%d",&A[i]);
     if ((A[i]<10)&&(A[i]>0))
       i++;
     else
       printf("I numeri devono essere positivi e minori di 10\n");
    }
  i=0;
  while (i<N)
    {printf("Elemento n.%d del secondo vettore : ",i);
     scanf("%d",&B[i]);
     if ((B[i]<10)&&(B[i]>0))
       i++;
     else
       printf("I numeri devono essere positivi e minori di 10\n");
    }

  for (i=0;i<N;i++)
    SOMMA[i]=sommatoria(A[i],B[i]);

  printf("\nVettore SOMMA\n");
  for (i=0;i<N;i++)
    printf("\t%d\n",SOMMA[i]);
}
```

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO B**

Si scriva un programma C che

1. legga da input due sequenze di massimo 4 numeri interi positivi (controllare che siano positivi e scartare i negativi). L'utente può terminare prima di avere inserito 4 numeri inserendo uno 0. I numeri letti vanno memorizzati in due vettori V1 e V2.
2. Crei un vettore di float MED che contiene le medie tra i quadrati dei corrispondenti valori nei vettori V1 e V2. Si noti i due vettori non contengono necessariamente lo stesso numero di elementi  
Per fare questo si usino le funzioni

```
float media(int A, int B);  
int quadrato(int A);
```

3. Stampi a video il contenuto di MED (solo i valori significativi)

ESEMPIO:

Numeri inseriti

7

3

-4 → scartato

6

0

6

3

1

2

I vettori V1 e V2 saranno:

V1	7	3	6	
V2	6	3	1	2

MED	42.5	9.0	18.5	
-----	------	-----	------	--

**COMPITO B**

```
#include <stdio.h>  
#define N 4  
  
int quadrato(int A)  
{return A*A;  
}  
  
float media(int A, int B)  
{return (quadrato(A)+quadrato(B))/2.0;  
}  
  
void main()  
{int V1[N],V2[N],i=0,j=0;  
float MED[N];  
do  
{printf("Elemento n.%d del primo vettore : ",i);  
scanf("%d",&V1[i]);  
if (V1[i]>=0)  
i++;  
else  
printf("I numeri devono essere positivi\n");  
}  
while ((i<N)&&(V1[i-1]!=0));  
if (V1[i-1]==0) i--;  
  
do  
{printf("Elemento n.%d del secondo vettore : ",j);  
scanf("%d",&V2[j]);  
if (V2[j]>=0)  
j++;  
else  
printf("I numeri devono essere positivi\n");  
}  
while ((j<N)&&(V2[j-1]!=0));  
if (V2[j-1]==0) j--;  
if (i>j) i=j;  
  
for (j=0;j<i;j++)  
MED[j]=media(V1[j],V2[j]);  
  
printf("\nVettore MED\n");  
for (j=0;j<i;j++)  
printf("\t%f\n",MED[j]);  
}
```

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO C**

Si scriva un programma C che

1. legga da input due sequenze di 5 interi positivi (si controlli che siano positivi e si scartino i negativi). I numeri letti vanno memorizzati in due vettori A e B.
2. Crei un vettore MAGG che contiene gli elementi di A che sono maggiori di tutti gli elementi di B. A tal fine si usi una funzione

```
int magg_di_tutti(int V[], int Dim, int E1);
```

che restituisce 1 se l'elemento E1 è maggiore di tutti gli elementi del vettore v di dimensione Dim, 0 altrimenti.

3. Stampi a video il contenuto di MAGG

ESEMPIO:

Numeri inseriti

```
7
3
6
-4 → scartato
4
2

5
3
4
2
5
```

I vettori A e B saranno:

A	7	3	6	4	2
B	5	3	4	2	5

MAGG	7	6			
------	---	---	--	--	--

**COMPITO C**

```
#include <stdio.h>
#define N 5

int magg_di_tutti(int V[], int Dim, int E1)
{int i;
 for (i=0;i<Dim;i++)
   if (E1<=V[i]) return 0;
 return 1;
}

void main()
{int A[N],B[N],MAGG[N],i=0,j=0;
 while (i<N)
   {printf("Elemento n.%d del primo vettore : ",i);
    scanf("%d",&A[i]);
    if (A[i]>0)
      i++;
    else
      printf("I numeri devono essere positivi\n");
   }
 i=0;
 while (i<N)
   {printf("Elemento n.%d del secondo vettore : ",i);
    scanf("%d",&B[i]);
    if (B[i]>0)
      i++;
    else
      printf("I numeri devono essere positivi\n");
   }

 for (i=0;i<N;i++)
   if (magg_di_tutti(B, N, A[i])) MAGG[j++]=A[i];

 printf("\nVettore MAGG\n");
 for (i=0;i<j;i++)
   printf("\t%d\n",MAGG[i]);
}
```

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO D**

Si scriva un programma C che

1. legga da input due sequenze di 5 interi positivi (si controlli che siano positivi e si scartino i negativi). I numeri letti vanno memorizzati in due vettori P1 e P2.
2. Crei un vettore MIN che contiene gli elementi di P2 minori di tutti gli elementi di P1. A tal fine si usi una funzione

```
int min_di_tutti(int E1, int V[], int Dim);
```

che restituisce 1 se l'elemento E1 è minore di tutti gli elementi del vettore v di dimensione Dim, 0 altrimenti.

3. Stampi a video il contenuto di MIN

ESEMPIO:

Numeri inseriti

7  
4  
6  
-4 → scartato  
4  
10

5  
3  
4  
2  
5

I vettori P1 e P2 saranno:

<b>P1</b>	7	4	6	4	10
<b>P2</b>	5	3	4	2	5

<b>MIN</b>	3	2			
------------	---	---	--	--	--

**COMPITO D**

```
#include <stdio.h>
#define N 5

int min_di_tutti(int E1, int V[], int Dim)
{int i;
 for (i=0;i<Dim;i++)
   if (E1>=V[i]) return 0;
 return 1;
}

void main()
{int P1[N],P2[N],MIN[N],i=0,j=0;
 while (i<N)
   {printf("Elemento n.%d del primo vettore : ",i);
    scanf("%d",&P1[i]);
    if (P1[i]>0)
      i++;
    else
      printf("I numeri devono essere positivi\n");
   }
 i=0;
 while (i<N)
   {printf("Elemento n.%d del secondo vettore : ",i);
    scanf("%d",&P2[i]);
    if (P2[i]>0)
      i++;
    else
      printf("I numeri devono essere positivi\n");
   }

 for (i=0;i<N;i++)
   if (min_di_tutti(P2[i], P1, N)) MIN[j++]=P2[i];

 printf("\nVettore MIN\n");
 for (i=0;i<j;i++)
   printf("\t%d\n",MIN[i]);
}
```

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO E**

Si scriva un programma C che

1. legga da input due sequenze di 5 interi positivi. La prima sequenza è composta di numeri pari (si controlli che siano pari e si scartino i dispari), mentre la seconda è composta da numeri dispari (si controlli che siano dispari e si scartino i pari). I numeri letti vanno memorizzati in due vettori P1 e P2. A tal fine si usi una funzione

```
int pari(int E1);
```

che restituisce 1 se l'elemento E1 è pari, 0 altrimenti.

2. Scambi gli elementi di P1 e P2 in modo che il primo di P1 sia scambiato con l'ultimo di P2, il penultimo di P1 con il secondo di P2 e così' via.
3. Stampi a video il contenuto di P1 e P2 modificati

ESEMPIO:

Numeri inseriti

8  
6  
6  
5 → scartato  
4  
2

3  
5  
7  
2 → scartato  
1  
9

I vettori P1 e P2 saranno:

<b>P1</b>	8	6	6	4	2
-----------	---	---	---	---	---

<b>P1</b>	9	1	7	5	3
-----------	---	---	---	---	---

<b>P2</b>	3	5	7	1	9
-----------	---	---	---	---	---

<b>P2</b>	2	4	6	6	8
-----------	---	---	---	---	---

**COMPITO E**

```
#include <stdio.h>
#define N 5

int pari(int E1)
{return (!(E1%2));
}

void main()
{int P1[N],P2[N],i=0,swap;
 while (i<N)
  {printf("Elemento n.%d del primo vettore : ",i);
   scanf("%d",&P1[i]);
   if ((pari(P1[i]))&&(P1[i]>0))
    i++;
   else
    printf("I numeri devono essere positivi e pari\n");
  }
 i=0;
 while (i<N)
  {printf("Elemento n.%d del secondo vettore : ",i);
   scanf("%d",&P2[i]);
   if (!(pari(P2[i]))&&(P2[i]>0))
    i++;
   else
    printf("I numeri devono essere positivi e dispari\n");
  }

 for (i=0;i<N;i++)
  {swap=P1[i];
   P1[i]=P2[N-1-i];
   P2[N-1-i]=swap;
  }

 printf("\nVettore P1\tVettore P2\n");
 for (i=0;i<N;i++)
  printf("\t%d\t\t%d\n",P1[i],P2[i]);
}
```

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO F**

Si scriva un programma C che

1. legga da input due sequenze di 6 interi positivi. La prima sequenza è composta di numeri pari (si controlli che siano pari e si scartino i dispari), mentre la seconda è composta da numeri dispari (si controlli che siano dispari e si scartino i pari). I numeri letti vanno memorizzati in due vettori P1 e P2. A tal fine si usi una funzione

```
int pari(int E1);
```

che restituisce 1 se l'elemento E1 è pari, 0 altrimenti.

2. Scambi gli elementi di P1 e P2 in modo che il primo di P1 sia scambiato con il secondo di P2 e il primo di P2 con il secondo di P1, il terzo di P1 con il quarto di P2 e il terzo di P2 con il quarto di P1 ecc..
3. Stampi a video il contenuto di P1 e P2 modificati

ESEMPIO:

Numeri inseriti

```
8
6
6
5 → scartato
4
2
10
3
5
7
2 → scartato
1
9
11
```

I vettori P1 e P2 saranno:

<b>P1</b>	8	6	6	4	2	10	<b>P1</b>	5	3	1	7	11	9
<b>P2</b>	3	5	7	1	9	11	<b>P2</b>	6	8	4	6	10	2

**COMPITO F**

```
#include <stdio.h>
#define N 6

int pari(int E1)
{return (!(E1%2));
}

void main()
{int P1[N],P2[N],i=0,swap;
 while (i<N)
  {printf("Elemento n.%d del primo vettore : ",i);
   scanf("%d",&P1[i]);
   if ((pari(P1[i]))&&(P1[i]>0))
    i++;
   else
    printf("I numeri devono essere positivi e pari\n");
  }
 i=0;
 while (i<N)
  {printf("Elemento n.%d del secondo vettore : ",i);
   scanf("%d",&P2[i]);
   if (!(pari(P2[i]))&&(P2[i]>0))
    i++;
   else
    printf("I numeri devono essere positivi e dispari\n");
  }

 for (i=0;i<N;i=i+2)
  {swap=P1[i];
   P1[i]=P2[i+1];
   P2[i+1]=swap;
   swap=P1[i+1];
   P1[i+1]=P2[i];
   P2[i]=swap;
  }

 printf("\nVettore P1\tVettore P2\n");
 for (i=0;i<N;i++)
  printf("\t%d\t\t%d\n",P1[i],P2[i]);
}
```

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO G**

Si scriva un programma C che

1. legga da input una sequenza di 4 interi positivi e una di 8 interi positivi (si controlli che siano positivi e si scartino i negativi). I numeri letti vanno memorizzati in due vettori P1 e P2.

2. Verifichi che tutti gli elementi di P1 siano contenuti in P2. A tal fine si usi una funzione

```
int contiene(int V1[], int Dim1, int V2[], int Dim2);
```

che restituisce 1 se tutti gli elementi del vettore V1 di dimensione Dim1 sono contenuti nel vettore V2 di dimensione Dim2, 0 altrimenti.

3. Stampi a video il contenuto dei due vettori e VERO o FALSO a seconda che tutti gli elementi di P1 siano contenuti o meno in P2.

ESEMPIO:

Numeri inseriti

```
5
3
6
-5 → scartato
7
```

```
2
5
9
6
8
-5 → scartato
1
7
3
```

I vettori P1 e P2 saranno:

**P1**

5	3	6	7
---	---	---	---

**P2**

2	5	9	6	8	1	7	3
---	---	---	---	---	---	---	---

E il risultato sarà: VERO

**COMPITO G**

```
#include <stdio.h>
#define N 4

int contiene(int V1[], int Dim1, int V2[], int Dim2)
{int i=0,j=0,trovato=1;
 while ((i<Dim1)&&(trovato))
 {trovato=0;
  while ((j<Dim2)&&(!trovato))
   if (V1[i]==V2[j])
    {j=0;
     i++;
     trovato=1;
    }
   else j++;
 }
 return trovato;
}

void main()
{int P1[N],P2[2*N],i=0;
 while (i<N)
 {printf("Elemento n.%d del primo vettore : ",i);
  scanf("%d",&P1[i]);
  if (P1[i]>0) i++;
  else printf("I numeri devono essere positivi\n");
 }
 i=0;
 while (i<2*N)
 {printf("Elemento n.%d del secondo vettore : ",i);
  scanf("%d",&P2[i]);
  if (P2[i]>0) i++;
  else printf("I numeri devono essere positivi\n");
 }

 printf("\nVettore P1\tVettore P2\n");
 for (i=0;i<N;i++)
  printf("\t%d\t\t%d\n",P1[i],P2[i]);
 for (i=N;i<2*N;i++)
  printf("\t\t\t\t%d\n",P2[i]);

 if (contiene(P1,N,P2,2*N)) printf("\n\tVERO");
 else printf("\n\tFALSO");
}
```

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 27 NOVEMBRE 2003**  
**COMPITO H**

Si scriva un programma C che

1. legga da input una sequenza di 4 interi positivi e una di 8 interi positivi (si controlli che siano positivi e si scartino i negativi. I numeri letti vanno memorizzati in due vettori P1 e P2.
2. Crei un vettore OCCURR che contiene in ogni elemento OCCURR[i] il numero di volte in cui l'elemento P1[i] compare in P2. A tal fine si usi una funzione

```
int occorrenze(int V[], int Dim, int E1);
```

che restituisce il numero di volte in cui **E1** compare nel vettore **V** di dimensione **Dim**.

3. Stampi a video il contenuto di OCCURR.

ESEMPIO:

Numeri inseriti

8  
7  
2  
-5 → scartato  
4

8  
4  
7  
-2 → scartato  
7  
7  
8  
7  
8

I vettori P1 e P2 saranno:

<b>P1</b>	8	7	2	4				
<b>P2</b>	8	4	7	7	7	8	7	8

<b>OCCURR</b>	3	4	0	1
---------------	---	---	---	---

**COMPITO H**

```
#include <stdio.h>
#define N 4

int occorrenze(int V[], int Dim, int E1)
{int i,occ=0;
  for (i=0;i<Dim;i++)
    if (V[i]==E1) occ++;
  return occ;
}

void main()
{int P1[N],P2[2*N],OCCURR[N],i=0;
  while (i<N)
    {printf("Elemento n.%d del primo vettore : ",i);
     scanf("%d",&P1[i]);
     if (P1[i]>0)
       i++;
     else
       printf("I numeri devono essere positivi\n");
    }
  i=0;
  while (i<2*N)
    {printf("Elemento n.%d del secondo vettore : ",i);
     scanf("%d",&P2[i]);
     if (P2[i]>0)
       i++;
     else
       printf("I numeri devono essere positivi\n");
    }

  for (i=0;i<N;i++)
    OCCURR[i]=occorrenze(P2,2*N,P1[i]);

  printf("\nVettore OCCURR\n");
  for (i=0;i<N;i++)
    printf("\t%d\n",OCCURR[i]);
}
```