

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 3 DICEMBRE 2002**  
**COMPITO A**

Si scriva un programma C che

1. chieda all'utente di inserire una sequenza di interi di lunghezza 8. Inserisca i numeri letti alternativamente in 2 vettori VETT1 e VETT2 iniziando da VETT1.
2. crei un vettore VETT3 che contiene i valori di VETT1 che sono maggiori in valore assoluto dei corrispondenti di VETT2. Per il valore assoluto si scriva una funzione

```
int val_ass(int N1);  
che restituisce il valore assoluto di N1.
```

3. stampare gli elementi di VETT3 positivi

ESEMPIO:

Numeri inseriti

-2  
4  
6  
-3  
3  
-2  
7  
-9

I vettori VETT1 e VETT2 saranno:

**VETT1**

-2	6	-3	7
----	---	----	---

**VETT2**

4	-3	-2	-9
---	----	----	----

**VETT3**

6	-3		
---	----	--	--

Il programma stamperà

6

infatti -2 ha valore assoluto minore di 4 e non va inserito in VETT3, 6 e -3 hanno valore assoluto maggiore rispettivamente di 3 e 2 e 7 ha valore assoluto minore di 9. VETT3 quindi conterrà solo 6 e -3. Stampando l'unico positivo si ottiene 6.

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 3 DICEMBRE 2002**  
**COMPITO B**

Si scriva un programma C che

1. legga da input due sequenze di esattamente 5 interi. I numeri letti vanno memorizzati in due vettori P1 e P2 relativi ai punteggi di una partita di carte su 5 incontri.
2. conti il numero di volte in cui vince la partita il primo giocatore e il numero di volte in cui vince il secondo. A tale scopo si usi una funzione

```
int winner(int G1, int G2);
```

che restituisce **1** se **G1 > G2**, **2** se **G2 > G1**. (il pareggio non è contemplato).

3. stampi a video il vincitore (inteso come giocatore 1 o giocatore 2) e il risultato dell'intera partita. Si sfrutti anche per questo punto la funzione vince sviluppata al punto precedente.

ESEMPIO:

Numeri inseriti

7  
3  
6  
4  
6

6  
6  
4  
6  
7

I vettori P1 e P2 saranno:

<b>P1</b>	<b>7</b>	<b>3</b>	<b>6</b>	<b>4</b>	<b>6</b>
<b>P2</b>	<b>6</b>	<b>6</b>	<b>4</b>	<b>6</b>	<b>7</b>

Il numero di partite vinte dal primo giocatore è 2, mentre quello del secondo giocatore è 3.  
Il programma stamperà

GIOCATORE 2  
Punteggio: 3-2

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO - 3 DICEMBRE 2002**  
**COMPITO C**

Si scriva un programma C che

3. legga da input due sequenze di 4 interi minori o uguali a 4 (si controlli che ogni numero siano minore o uguale a 4, in caso contrario lo si scarti). I numeri letti vanno memorizzati in due vettori B e EXP
4. crei un vettore POTENZE contenente le potenze degli elementi contenuti in B e i corrispondenti (stessa posizione) del vettore EXP. A tale scopo di definisca una funzione

```
int pow(int B, int E);
```

che calcola la potenza del numero B elevato a E come sequenza di prodotti.

3. stampi il vettore POTENZA

ESEMPIO:

Numeri inseriti

2  
3  
1  
4

3  
2  
2  
1

I vettori B e EXP saranno:

<b>B</b>	2	3	1	4
----------	---	---	---	---

<b>EXP</b>	3	2	2	1
------------	---	---	---	---

Il programma creerà e stamperà

<b>POTENZA</b>	8	9	1	4
----------------	---	---	---	---

**PROVA PRATICA DI FONDAMENTI DI INFORMATICA L-A**  
**Prof. MICHELA MILANO – 3 DICEMBRE 2002**  
**COMPITO D**

Si scriva un programma C che

1. legga da input due sequenze di 4 interi tutti divisibili per 2 e li memorizzi in due vettori P e S. Si controlli che i numeri siano divisibili per 2 tramite la funzione

```
int div_due(int N1);
```

che restituisce **1** se **N1** è divisibile per 2, **0** altrimenti.

2. Si crei un terzo vettore PROD tale che **PROD[i] = P[i]\*S[i]**
3. Calcoli e stampi la somma dei valori di **PROD[i]**

$$\sum_{i=1}^4 \text{PROD}[i]$$

ESEMPIO:

Numeri inseriti

4  
2  
8  
2

6  
4  
8  
2

I vettori P e S saranno:

<b>P</b>	<table border="1"><tr><td>4</td><td>2</td><td>8</td><td>2</td></tr></table>	4	2	8	2
4	2	8	2		
<b>S</b>	<table border="1"><tr><td>6</td><td>4</td><td>8</td><td>2</td></tr></table>	6	4	8	2
6	4	8	2		

<b>PROD</b>	<table border="1"><tr><td>24</td><td>8</td><td>64</td><td>4</td></tr></table>	24	8	64	4
24	8	64	4		

Il programma calcolerà e stamperà

## COMPITO A

```
#define MAX 8

int val_ass(int N)
{if (N<0) return -N;
  else return N;
}

void main()
{
  int vett1[MAX], vett2[MAX], vett3[MAX];
  int i, k, num;

  for(i=0; i< MAX/2; i++) {
    printf("inserire un numero");
    scanf("%d", &vett1[i]);
    printf("inserire un numero");
    scanf("%d", &vett2[i]);
  }

  k=0;
  for(i=0; i< MAX/2; i++)
  if (val_ass(vett1[i]) > val_ass(vett2[i]))
    {vett3[k]=vett1[i];
     k++;}
  for(i=0; i< k; i++)
    if(vett3[i]>0)
      printf("%d ", vett3[i]);
}
```

## COMPITO B

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

int vince(int G1, int G2)
{if (G1>G2) return 1;
  else return 2;
}

void main()
{int P1[N], P2[N];
  int i, VINTE1=0, VINTE2=0, Vincitore;

  for (i = 0; i < N; i++)
    {printf("punteggio del set %d primo giocatore\n ",i);
      scanf("%d", &P1[i]);
    }
  for (i = 0; i < N; i++)
    {printf("punteggio del set %d secondo giocatore\n ",i);
      scanf("%d", &P2[i]);
    }

  for (i = 0; i < N; i++)
    {if (vince(P1[i],P2[i])==1)
        VINTE1++;
      else VINTE2++;
    }
  Vincitore = vince(VINTE1, VINTE2);
  printf("Vincitore: giocatore %d\n", Vincitore);
  printf("Punteggio: %d-%d\n", VINTE1, VINTE2);
}
```

## COMPITO C

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

int pow(int N1, int N2)
{int P=1,i;
  for(i=0; i<N2; i++)
    P=P*N1;
  return P;
}

void main()
{int B[MAX], EXP[MAX], POTENZA[MAX];
  int i;

  for (i=0; i<MAX; i++)
  do
  {printf("inserisci un numero minore o uguale a 4: \n");
   scanf("%d",&B[i]);
  } while (B[i] > 4);

  for (i=0; i<MAX; i++)
  do
  {printf("inserisci un numero minore di 5: \n");
   scanf("%d",&EXP[i]);
  } while (EXP[i]>= 5);

  for(i=0; i<MAX; i++)
  { POTENZA[i] = pow(B [i],EXP[i]);
    printf("%d  ", POTENZA[i]);
  }
  printf("\n");

}
```

## COMPITO D

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

int div_due(int N)
{if (N % 2 == 0) return 1;
  else return 0;
}

void main()
{int P[MAX], S[MAX], PROD[MAX];
  int i, SOMMA=0;

  printf("prima sequenza \n");
  for (i=0; i<MAX; i++)
  do
  {printf("inserisci un numero divisibile per 2: \n");
    scanf("%d",&P[i]);
  } while (div_due(P[i])!=0);

  printf("seconda sequenza \n");
  for (i=0; i<MAX; i++)
  do
  {printf("inserisci un numero divisibile per 2: \n");
    scanf("%d",&S[i]);
  } while (div_due(S[i])!=0);

  for (i=0; i<MAX; i++)
    PROD[i]= P[i]*S[i];

  for (i=0; i<MAX; i++)
    SOMMA=SOMMA+PROD[i];

  printf("Somma di prodotti: %d \n", SOMMA);
}
```