

ESERCIZIO 1

Si scriva un programma che prenda in ingresso i coefficienti a , b , c di un'equazione di secondo grado e che restituisca i risultati dell'equazione.

A tal fine si scriva una funzione

```
int delta_pos(float a, float b, float c);
```

che dati a , b , c controlli se il delta dell'equazione è positivo, e una funzione

```
float delta(float a, float b, float c);
```

che dati a , b , c restituisca il delta dell'equazione.

Si organizzi poi il progetto su più file, ponendo le funzioni (o le procedure) in un file sorgente diverso da quello contenente il main.

ESERCIZIO 1

FILE esercizio1.c

```
#include <stdio.h>
#include <math.h>
#include "funzionil.h"

main()
{float a, b, c;
 double x1, x2;
 printf("Inserisci i tre coefficienti: ");
 scanf("%f %f %f", &a, &b, &c);
 if ((a!=0)&&(esiste_delta(a,b,c)))
 {x1=(-b + sqrt(delta(a,b,c)))/(2*a);
  x2=(-b - sqrt(delta(a,b,c)))/(2*a);
  printf("I risultati sono x1=%g, x2=%g.",
        x1, x2);
 }
 else printf("Equazione non risolvibile");
}
```

ESERCIZIO 1

FILE funzioni1.h

```
int delta_pos(float a, float b, float c);  
float delta(float a, float b, float c);
```

FILE funzioni1.c

```
int delta_pos(float a, float b, float c)  
{ return ((b * b) >= (4 * a * c));  
}  
  
float delta(float a, float b, float c)  
{ return ((b * b) - (4 * a * c));  
}
```

ESERCIZIO 2

Si scriva una procedura:

```
void leggi_num (int N, int* Resto);
```

che chieda all'utente N numeri interi e restituisca nel secondo parametro il resto della divisione intera tra il maggior numero inserito e N. (Se, per esempio, N=4 e il maggior numero inserito è 9, la funzione deve ritornare il valore 1).

Si scriva un possibile main che chiami questa funzione.

Si organizzi poi il progetto su più file, ponendo le funzioni (o le procedure) in un file sorgente diverso da quello contenente il main.

ESERCIZIO 2

FILE esercizio2.c

```
#include <stdio.h>
#include "procedura2.h"

main()
{int N, Resto;
  printf("Inserisci il numero N: ");
  scanf("%d", &N);
  leggi_num(N, &Resto);
  printf("Il risultato e' %d\n", Resto);
}
```

ESERCIZIO 2

FILE procedura2.h

```
void leggi_num (int N, int* Resto);
```

FILE procedura2.c

```
void leggi_num (int N, int* Resto)
{int max, n, i;
  for (i=0; i<N; i++)
    {printf("Inserisci il %d° numero: ",i+1);
      scanf("%d",&n);
      if ((n>max) || (i==0)) max=n;
    }
  *Resto= max % N;
}
```

ESERCIZIO 3

Si scriva un programma che prenda in ingresso due numeri interi **a** e **b** e restituisca la loro somma se sono primi fra di loro, altrimenti il loro prodotto.

A tal fine si scriva e si utilizzi una procedura:

```
void MassComDiv(int a, int b, int* mcd);
```

che dati **a** e **b** restituisca nel terzo parametro il loro Massimo Comune Divisore.

Si organizzi poi il progetto su più file, ponendo le funzioni (o le procedure) in un file sorgente diverso da quello contenente il main.

ESERCIZIO 3

FILE esercizio3.c

```
#include <stdio.h>
#include "procedura3.h"

main()
{int a, b, ris;
  printf("Inserisci i due numeri: ");
  scanf("%d %d", &a, &b);
  MCD(a, b, &ris);
  printf("\nIl risultato e' %d\n",
        (ris == 1) ? a + b : a * b);
}
```

ESERCIZIO 3

FILE procedura3.h

```
int MCD(int a, int b, int* mcd);
```

FILE procedura3.c

```
int MCD(int a, int b, int* mcd)
{if (a == b) *mcd = a;
  else if (a > b)
    MCD(a-b, b, mcd);
    else MCD(b-a, a, mcd);
}
```

ESERCIZIO 4

Si scriva un programma che prenda in ingresso un numero N compreso tra 1 e 20 e restituisca N^N se il numero è ≤ 10 , oppure $\prod_{i=1}^N i$ se il numero è > 10 .

A tal fine si scriva una funzione

```
double potenza(int x);
```

che dato x calcoli x^x usando come operazione primitiva il prodotto, e una funzione

```
double prodotto(int x);
```

che dato x calcoli $\prod_{i=1}^x i$

Si organizzi poi il progetto su più file, ponendo le funzioni (o le procedure) in un file sorgente diverso da quello contenente il main.

ESERCIZIO 4

FILE esercizio4.c

```
#include <stdio.h>
#include "funzioni4.h"

main()
{int N;
 double result;
 do
 {printf("Inserisci un numero intero
        compreso tra 1 e 20: ");
  scanf("%d", &N);}
 while ((N<1)|| (N>20));

 result= (N <= 10)? potenza(N): prodotto(N);
 printf("Il risultato e' %lf", result);
}
```

ESERCIZIO 4

```
double potenza(int x);
double prodotto(int x);
```

FILE funzioni4.h

```
double potenza(int x)
{ int i;
  double P=1;
  for(i=1; i<=x; i++)
    P = P * x;
  return P;
}
```

FILE funzioni4.c

```
double prodotto(int x)
{ int i;
  double P=1;
  for (i=1; i<=x; i++)
    P = P * i;
  return P;
}
```