

SOLUZIONE 1

Scrivere un programma C che, dato un numero intero N, calcoli il fattoriale di tutti i numeri interi minori o uguali a N. Bisogna prima scrivere il codice per calcolare il fattoriale di un numero intero k, e poi inserirlo all'interno di un ciclo da 2 a N.

```
main()
{ int N,f,i,k;
printf("Inserire N: ");
scanf("%d",&N);
for (k=2; k<=N; k++)
/*CICLO ESTERNO, DA 2 A N */
{ /* si calcola il fattoriale di k */
f=1;
for (i=2; i<=k; i++) f = f*i;
printf("Fatt(%d)=%d\n",k,f);
}
}
```

SOLUZIONE 1

Variante: Osservando che $\text{fatt}(N) = \text{fatt}(N-1) * N$, è possibile utilizzare un ciclo solo.

```
#include <stdio.h>
main()
{
int N,F,I,k;
printf("Inserire N: ");
scanf("%d",&N);
F=1;
for (k=2; k<=N; k++)
{
F = F*k;
printf("Fatt(%d)=%d\n",k,F);
}
}
```

SOLUZIONE 2

Dato un numero reale N e uno intero M, calcolare la potenza di N alla M con $M \geq 0$.

```
#include <stdio.h>
main()
{
    double N,pot=1;
    int M,i;
    printf("Inserire N e M (M>=0): ");
    scanf("%lf%d",&N,&M);
    for (i=1;i<=M;i++)
        pot = pot*N;
    printf("N^M = %lf",pot);
}
```

SOLUZIONE 2

Variante 1: ciclo while

```
#include <stdio.h>
main()
{
    double N,pot=1;
    int M,i;
    printf("Inserire N e M (M>=0): ");
    scanf("%lf%d",&N,&M);
    i=1;
    while(i<=M)
    {
        pot = pot*N;
        i++;
    }
    printf("N^M = %lf",pot);
}
```

SOLUZIONE 2

Variante 2: Supporre che M possa assumere valori negativi.

```
main()
{
    double N, pot=1;
    int M, i;
    printf("Inserire N e M (M>=0): ");
    scanf("%lf%d",&N,&M);

    if (M>=0)
        for (i=1;i<=M;i++) pot = pot*N;
    else {
        for (i=1;i<=(-M);i++)
            pot = pot/N;
    }
    printf("N^M = %lf",pot);
}
```

SOLUZIONE 3

- Scrivere un programma C che calcoli la somma di due numeri tramite i cicli iterativi utilizzando solo l'operazione di incremento

```
main()
{ int i;
  int x,y,somma;
  printf("digita due numeri interi\n");
  scanf("%d%d", &x, &y);
  somma=x;
  for(i=0; i<y;i++)
      somma++;
  printf("la somma di %d e %d e' %d\n",
    x,y,somma);
}
```

SOLUZIONE 4

Dato un intero R disegnare (su R righe) la figura:

```
*
***
*****
*****
*****
```

```
void main() {
int i,r,riga,spazi,stelle;
printf("Inserire r: ");
scanf("%d",&r);

spazi=r-1;
stelle=1;
for (riga=1;riga<=r;riga++){
for (i=1;i<=spazi;i++) printf(" ");
for (i=1;i<=stelle;i++) printf("*");
printf("\n");
spazi--;
stelle++;
stelle++;
}
}
```

SOLUZIONE 5

Calcolare iterativamente:

$$\pi = 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots \right)$$

```
void main() {
int i=0;
double pi=0;

while(i<1000000){
pi=pi+1/(2*i+1.0);
i++;

pi=pi-1/(2*i+1.0);
i++;
}

printf("%f\n",4*pi);
}
```

SOLUZIONE 6

Questo programma legge numeri interi, positivi e negativi, finché l'utente non digita il valore 0 (valore di terminazione). Quando l'utente ha finito di immettere gli interi, il programma stampa la media dei soli interi positivi.

Algoritmo:

Passo 1: Leggi un primo numero

Passo 2: Finché il numero letto è diverso da zero fai:

Passo 2-1: Se il numero è positivo aggiungilo a somma totale; inoltre incrementa di uno il conto dei numeri positivi letti.

Passo 2-2: Leggi un nuovo numero e ripeti il ciclo.

Passo 3: Se sono stati inseriti dei numeri positivi,

Allora calcola la media.

Altrimenti stampa un messaggio di errore.

```
#include <stdio.h>
void main()
{
    float sum, media;
    int number, positiveNumbers;
    sum = 0; positiveNumbers = 0;

    /* Passo 1 */
    printf("Introduci i numeri (0 per finire): ");
    scanf("%d", &number);

    /* Passo 2 */
    while(number != 0)
    {if (number > 0) /* Passo 2-1 */
        {
            sum = sum + number;
            positiveNumbers++;
        }
        scanf("%d", &number); /* Passo 2-2 */
    }
    /* Passo 3 */
    if (positiveNumbers > 0)
    {
        media = sum / positiveNumbers;
        printf("La media è: %f", media);
    }
    else printf("Non hai inserito nessun numero positivo!");
}
}
```

SOLUZIONE 7

Realizzare un programma C che, chiesto all'utente un numero intero pari a N , ricavi e stampi a video tutte le terne pitagoriche con i cateti minori o uguali ad N .

Una terna pitagorica è formata da tre numeri naturali (a, b, c) tali che $a^2 + b^2 = c^2$. Esempio 3,4,5 ($9+16=25$).

N.B. : Calcolare solo le terne in cui $a < b$.

```
#include <stdio.h>
void main() {
    int n, a, b, c;

    printf("Inserisci il valore di n: ");
    scanf("\n%d", &n);
    for(a=1; a<=n; a++) {
        for(b=a+1; b<=n ; b++) {
            for(c=(b+1); c<(a+b); c++) {

                if (a*a+b*b==c*c) printf("%d, %d, %d\n", a, b, c);
            }
        }
    }
}
```