

Esercizi di programmazione in C

Esercizio 1

Scrivere un programma in linguaggio **C** che legga da tastiera una sequenza di lunghezza ignota a priori di numeri interi positivi. Il programma, a partire dal primo numero introdotto, stampa ogni volta la media di tutti i numeri introdotti. Terminare quando il numero inserito è negativo.

Soluzione:

```
#include <stdio.h>

void main()
{
    int a, i, t;
    float r;

    t = 0;
    i = 0;
    printf ("Inserisci un numero: ");
    scanf("%d",&a);
    while (a >= 0) {
        i++;
        t += a;
        r = t/i;
        printf ("La media attuale è: %.3f\n", r);
        printf ("Inserisci un numero: ");
        scanf("%d",&a);
    }
    printf ("La media finale è: %.3f\n", r);
}
```

Esercizio 2

Si scriva un programma in linguaggio C che acquisisca da tastiera una parola (cioè una stringa di caratteri priva di separatori) e la stampi a video **se e solo se** tale parola è **palindroma**, ossia leggibile nello stesso modo da destra a sinistra e viceversa (es. OSSESSO).

Per determinare la lunghezza della parola si può utilizzare la funzione **strlen(s)** contenuta nel file header **<string.h>**.

Soluzione:

```
#include <stdio.h>
#include <string.h>

void main()
{
    char s[100];
    int i, flag;

    i = 0;
    scanf ("%s", s);
    flag = 1;
    while ((i < (strlen(s)/2)) && (flag == 1)) {
        if (s[i] != s[strlen(s)-i-1]) {
            flag = 0;
        }
        i++;
    }
    if (flag == 1) {
        printf ("La parola è palindroma.\n");
    } else {
        printf ("La parola non è palindroma.\n");
    }
}
```

Esercizio 3

Scrivere un programma in linguaggio **C** che legga da tastiera una sequenza di numeri positivi e ad ogni numero letto ne stampi la somma progressiva. Il programma termina quando si introduce un numero minore o uguale a zero.

Soluzione:

```
#include <stdio.h>

void main()
{
    int a, t;

    t = 0;
    printf ("Inserisci un numero: ");
    scanf ("%d", &a);
    while (a >= 0) {
        t += a;
        printf ("La somma progressiva attuale è: %d\n",
t);
        printf ("Inserisci un numero: ");
        scanf ("%d", &a);
    }
    printf ("La somma progressiva finale è: %d\n", t);
}
```

Esercizio 4

Si scriva un programma in linguaggio C che legga da tastiera un numero intero A, lo riduca ad un valore compreso tra 0 e 127 **mediante sottrazione ripetuta** di un adeguato numero di volte del valore 128 (non si utilizzi l'operatore modulo o and), lo interpreti come caratteri ASCII e lo stampi sul video.

Soluzione:

```
#include <stdio.h>

void main()
{
    int a;

    printf ("Inserisci un numero: ");
    scanf ("%d", &a);
    while (a >= 128) {
        a -= 128;
    }
    printf ("%c\n", a);
}
```

Esercizio 5

Si scriva un programma C che legga da tastiera 5 numeri e stampi a video il maggiore tra essi, la loro media e la radice quadrata della somma.

Si noti che per effettuare la radice quadrata esiste la funzione double **sqrt** (double) definita nel file di header: **<math.h>**

Soluzione:

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

void main( )
{
    int i, a, max, somma;
    float r;

    max = 0;
    for (i=0; i<5; i++) {
        printf ("Inserisci un numero: ");
        scanf ("%d", &a);
        if (max < a) {
            max = a;
        }
        somma += a;
    }
    r = somma / 5;
    printf ("Il valore massimo inserito è: %d\n", max);
    printf ("La radice quadrata della somma è: %.3f\n",
           sqrt(somma));
    printf ("La media è: %.3f\n", r);
}
```

Esercizio 6

Chiedere 2 num. interi (a,b) e disegnare un rettangolo di dimensioni a*b usando il carattere " * ".

Soluzione:

```
#include <stdio.h>

void main( )
{
    int a, b, i, j;

    printf ("Inserisci il lato a: ");
    scanf ("%d", &a);
    printf ("Inserisci il lato b: ");
    scanf ("%d", &b);
    for (i=0; i<a; i++) {
        printf ("*");
    }
    printf ("\n");
    for (i=0; i<b-2; i++) {
        printf ("*");
        for (j=0; j<a-2; j++) {
            printf (" ");
        }
        printf ("*\n");
    }
    for (i=0; i<a; i++) {
        printf ("*");
    }
    printf ("\n");
}
```

Esercizio 7

Generare un numero a caso e chiedere all'utente un numero fino a quando non e' uguale a quello generato casualmente. Dire ogni volta se il numero immesso e' > o < di quello iniziale.

Soluzione:

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    int a, t, r;

    r = rand();
    a = -1;
    while (a != r) {
        printf ("Inserisci un numero: ");
        scanf ("%d",&a);
        if (a != r) {
            if (a<r) {
                printf ("Inserisci un numero maggiore.\n");
            } else {
                printf ("Inserisci un numero minore.\n");
            }
        }
    }
}
```

1 Strutture di controllo

1.1 Multiplo

Testo

Legge due numeri e controlla se il primo è multiplo del secondo.

Implementazione

```
#include <stdio.h>
void main()
{
    unsigned int m, n;

    scanf ("%u", &m);
    scanf ("%u", &n);
    if (m % n == 0)
    {
        printf ("%u è multiplo di %u\n", m, n);
    }
    else
    {
        printf ("%u non è multiplo di %u\n", m, n);
    }
}
```

1.2 Massimo Comun Divisore

Testo

Legge due numeri e calcola l'MCD.

Implementazione

```
#include <stdio.h>
void main()
{
    unsigned int n, m, mcd, min, contatore;

    scanf ("%u", &n);
    scanf ("%u", &m);
    if (n <= m)
    {
        min = n;
    }
    else
    {
        min = m;
    }
    contatore = 1;
    while (contatore <= min)
    {
        if (n % contatore == 0 && m % contatore == 0)
```

```

    {
        mcd = contatore;
    }
    contatore = contatore + 1;
}
printf ("%u", mcd);
}

```

1.3 Valutazione triangolo

Testo

Legge tre interi e stabilisce se possono essere le lunghezze dei lati di un triangolo. In caso affermativo, trova la tipologia del triangolo.

Implementazione

```

#include <stdio.h>
void main()
{
    unsigned int x, y, z;

    scanf ("%u", &x);
    scanf ("%u", &y);
    scanf ("%u", &z);
    if ((x < y + z) && (y < x + z) && (z < x + y))
    {
        if (x == y && y == z)
        {
            printf ("Triangolo equilatero");
        }
        else
        {
            if (x == y || y == z || x == z)
            {
                printf ("Triangolo isoscele");
            }
            else
            {
                printf ("Triangolo scaleno");
            }
        }
    }
    else
    {
        printf ("Non è un triangolo");
    }
}

```

1.4 Calcolatrice

Testo

L'utente inserisce un carattere ('+', '-', '*', '/') che rappresenta l'operazione da eseguire e due operandi (numeri float). L'operazione viene eseguita, se possibile.

Implementazione

```
#include <stdio.h>
void main()
{
    float op1, op2, tot;
    char operazione;
    unsigned int errore = 0;

    printf ("Operazione (+,-,* , /): ");
    scanf ("%c", &operazione);
    printf ("Primo operando: ");
    scanf ("%f", &op1);
    printf ("Secondo operando: ");
    scanf ("%f", &op2);
    switch (operazione)
    {
        case '+': tot = op1 + op2;
                     break;
        case '-': tot = op1 - op2;
                     break;
        case '*': tot = op1 * op2;
                     break;
        case '/': if (op2 != 0)
                    {
                        tot = op1 / op2;
                    }
                    else
                    {
                        errore = 1;
                    }
                     break;
        default:   printf ("Operazione?\n");
                     errore = 1;
    }
    if (errore == 0)
    {
        printf ("Risultato: %f\n", tot);
    }
    else
    {
        printf ("Errore!\n");
    }
}
```

1.5 Anno bisestile

Testo

L'utente inserisce un anno ed il calcolatore verifica se è bisestile.

Se l'utente inserisce un numero minore di zero il programma termina (senza ovviamente fare alcuna verifica); altrimenti, al termine della verifica, si ricomincia daccapo.

Un anno è bisestile se è divisibile per 4 ma non per 100, oppure se è divisibile per 400.

Pseudocodice

```
[leggi l'anno]
while ([l'anno è un numero positivo])
{
    [calcola il resto della divisione intera dell'anno per 4, per 100 e per 400]
    if ([il primo resto è zero (cioè, l'anno è divisibile per 4) AND
          il secondo resto non è zero (cioè, l'anno NON è divisibile per 100),
          OR il terzo resto è zero (cioè, l'anno è divisibile per 400)])
    {
        [scrivi che l'anno è bisestile]
    }
    else
    {
        [scrivi che l'anno non è bisestile]
    }
    [leggi l'anno]
}
```

Implementazione

```
#include <stdio.h>
void main ()
{
    unsigned int restoDiv4, restoDiv100, restoDiv400;
    int anno;
    printf ("Inserire un anno: ");
    scanf ("%d", &anno);
    while (anno >= 0)
    {
        restoDiv4 = anno % 4;
        restoDiv100 = anno % 100;
        restoDiv400 = anno % 400;
        if (restoDiv4==0 && restoDiv100!=0 || restoDiv400==0)
        {
            printf("L'anno è bisestile\n");
        }
        else
        {
            printf ("L'anno non è bisestile\n");
        }
        printf ("Inserire un altro anno: ");
        scanf ("%d", &anno);
    }
}
```

1.6 Da minuscolo a maiuscolo

Testo

L'utente inserisce una sequenza di caratteri, terminata dal carattere '#'. Il programma visualizza i caratteri, trasformando tutte le lettere minuscole in maiuscole.

Eventuali caratteri diversi dalle lettere minuscole e dal '#' devono essere visualizzati come '_'; il carattere '#' deve essere anch'esso visualizzato.

Pseudocodice

```
do
{
    [leggi carattere]
    if ([carattere è compreso tra 'a' e 'z'])
    {
        [calcola e scrivi il carattere maiuscolo corrispondente, togliendo 32 al codice ASCII]
    }
    else if ([carattere non è '#'])
    {
        [scrivi '_']
    }
    else
    {
        [scrivi '#']
    }
} while ([carattere è diverso dal '#'])
```

Implementazione

```
#include <stdio.h>
void main()
{
    char carattere;
    printf ("Inserisci caratteri minuscoli; termina con #: ");
    do
    {
        scanf ("%c", &carattere);
        if (carattere >= 97 && carattere <= 122)
        {
            printf ("%c", carattere - 32);
        }
        else if (carattere != '#')
        {
            printf ("_");
        }
        else
        {
            printf ("#");
        }
    } while (carattere != '#');
}
```

1.7 Conversione temperature

Testo

L'utente inserisce una temperatura in Celsius e il calcolatore la converte in Fahrenheit ed in Kelvin.
Se la temperatura inserita è minore dello zero assoluto, il calcolatore segnala un errore.

Se l'utente inserisce una temperatura uguale allo zero assoluto il programma termina dopo la conversione, altrimenti si ricomincia daccapo.

Ricordiamo che:

$$\text{Fahrenheit} = (9/5) \cdot \text{Celsius} + 32$$

$$\text{Kelvin} = \text{Celsius} + 273,15$$

Pseudocodice

```
do
{
    [leggi la temperatura Celsius]
    if ([temperatura Celsius è maggiore o uguale a -273.15])
    {
        [converti la temperatura Celsius in Fahrenheit e Kelvin]
        [scrivi le temperature Fahrenheit e Kelvin]
    }
    else
    {
        [scrivi che c'è un errore: non esistono temperature Celsius minori di -273.15]
    }
} while ([temperatura Celsius è maggiore di -273.15])
```

Implementazione

```
#include <stdio.h>
void main()
{
    const float zeroAssolutoCelsius = -273.15;
    float celsius, fahrenheit, kelvin;

    do
    {
        printf ("Inserire temperat. in °C (-273.15 = fine): ");
        scanf ("%f", &celsius);
        if (celsius >= zeroAssolutoCelsius)
        {
            fahrenheit = (9.0 / 5.0) * celsius + 32;
            kelvin = celsius - zeroAssolutoCelsius;
            printf ("Fahrenheit: %f, Kelvin: %f\n", fahrenheit,
                   kelvin);
        }
        else
        {
            printf ("Errore!\n");
        }
    } while (celsius != zeroAssolutoCelsius);
}
```

1.8 Fattoriale

Testo

L'utente inserisce alcuni numeri naturali e l'elaboratore calcola il fattoriale per ognuno di essi.
Ricordiamo che i fattoriale è definito come:

$$\begin{array}{ll} n! = 1 & \text{sse } n = 0 \\ n! = n \cdot (n - 1)! & \text{sse } n > 0 \end{array}$$

ma può essere calcolato:

$$\begin{array}{ll} n! = 1 & \text{sse } n = 0 \\ n! = 1 \cdot 2 \cdot 3 \cdots n & \text{sse } n > 0 \end{array}$$

Pseudocodice

[leggi quanti numeri si vogliono inserire]
for ([esegui un numero di volte pari al valore letto al passo precedente])
 {
 [leggi numero naturale]
 [assegna 1 alla variabile che conterrà il fattoriale]
 for ([il contatore va da 1 al numero naturale letto, con passo 1])
 {
 [la variabile che conterrà il fattoriale viene moltiplicata
 per il valore corrente del contatore]
 }
 [scrivi la variabile che, adesso, contiene il fattoriale]
 }

Implementazione

```
#include <stdio.h>
void main()
{
    unsigned int n, i, max, j;
    unsigned long int fat;

    printf ("Quanti numeri?: ");
    scanf ("%u", &max);
    for (j = 1; j <= max; j++)
    {
        printf ("Introduci numero naturale: ");
        scanf ("%u", &n);
        fat = 1;
        for (i = 1; i <= n; i++)
        {
            fat = fat * i;
        }
        printf ("Il fattoriale è: %lu\n", fat);
    }
}
```