

# Esercizio 1

---

## Esercizio 1 – Triangoli

- Si scriva un programma, in linguaggio C, che legga da input le lunghezze dei lati di un triangolo e determini se il triangolo è equilatero, isoscele o scaleno

1

## Esercizio 1 - Soluzione

---

```
double primo, secondo, terzo;
printf("Lunghezze lati triangolo? ");
scanf("%lf%lf%lf", &primo, &secondo, &terzo);
if (primo == secondo)
{
    if (secondo == terzo)
        printf("Equilatero\n");
    else
        printf("Isoscele\n");
}
else
{
    if (secondo == terzo || primo == terzo)
        printf("Isoscele\n");
    else
        printf("Scaleno\n");
}
```

2

## Esercizio 2

---

### Esercizio 2 - Operazioni aritmetiche

- Realizzare un programma che, presi in input 2 operandi reali e un operatore (+, -, \*, /), esegue l'operazione stampandone il risultato
- Nel caso in cui l'operatore sia errato o l'operazione non eseguibile, si stampi "undefined"

3

## Esercizio 2 - Soluzione

---

```
#include <stdio.h>
int main ()
{
    double op1, op2;
    char type;
    scanf("%lf %c %lf", &op1, &type, &op2);
    switch(type)
    {
        case '+': printf("= %lf\n", op1 + op2); break;
        case '-': printf("= %lf\n", op1 - op2); break;
        case '*': printf("= %lf\n", op1 * op2); break;
        case '/':
            if (op2 != 0) printf("= %lf\n", op1 / op2);
            else printf("= undefined\n");
            break;
        default: printf("= undefined\n"); break;
    }
    return 0;
}
```

4

## Esercizio 3

---

### Esercizio - sequenze di cifre

- Realizzare un programma che prende in input una sequenza di cifre (tra 1 e 9) e calcola la somma massima fra le sotto-sequenze di cifre non decrescente
- Il programma termina quando viene inserito lo 0
- Esempio:

2	2	4	5	3	9	3	1	5	0
<b>13</b>				12		3	6		

5

## Esercizio 3 - Soluzione

---

```
char cifra;
int curVal = 0, oldVal = 0, somma = 0, sommaMax = 0;
do
{
    scanf("%c", &cifra);
    getchar();
    curVal = cifra - '0';
    if (curVal < oldVal)
    {
        if (somma > sommaMax)
            sommaMax = somma;
        somma = curVal;
    }
    else
        somma = somma + curVal;
    oldVal = curVal;
}
while (curVal != 0);
printf("Massima somma trovata: %d\n", sommaMax);
```

6

## Esercizio 4

---

### Esercizio 4 - ciclo per il calcolo del massimo e del minimo

- Realizzare un programma che calcola il minimo e il massimo di una serie di valori
- Il numero di valori deve essere costante e definito tramite una opportuna *costante simbolica*
- Se la differenza tra il massimo ed il minimo supera 10, il programma termina, altrimenti aspetta una nuova serie di valori
- Incapsulare il calcolo del minimo e del massimo in una funzione

7

## Esercizio 4 - Soluzione (1)

---

```
#include <stdio.h>

int max(int v1, int v2)
{
    if(v1 > v2)
        return v1;
    else
        return v2;
}

int min(int v1, int v2)
{
    return v1 < v2 ? v1 : v2;
}

...
```

8

## Esercizio 4 - Soluzione (2)

---

```
#include <stdio.h>
#define MAX_REQUEST 5
int main ()
{
    int curValue, maxValue, minValue, index;
    do
    {
        for(index = 0; index < MAX_REQUEST; index++)
        {
            printf("Inserire il valore %d: ", index+1);
            scanf("%d", &curValue);
            if(index == 0) //inizializzo max e min
            {
                maxValue = curValue;
                minValue = curValue;
            }
            else
            {
                maxValue = max(maxValue, curValue);
                minValue = min(minValue, curValue);
            }
        }
        printf("Calcolati: max = %d, min = %d\n", maxValue, minValue);
    } while(maxValue - minValue <= 10);
    return 0;
}
```

9

## Esercizio 5

---

### Esercizio 5 - Calcolo del mcm tra numeri interi

- Realizzare un programma che prende in input una serie di numeri interi, calcolando via via il minimo comune multiplo tra essi; il programma deve terminare quando il mcm diventa più grande di 100

## Esercizio 5 - Soluzione (1)

---

```
int mcd(int a, int b)
{
    int m, n;
    m = a;
    n = b;
    while(m != n)
    {
        if(m > n)
            m = m - n;
        else
            n = n - m;
    }
    return m;
}

int mcm(int a, int b)
{
    return (a * b) / mcd(a, b);
}
```

11

## Esercizio 5 - Soluzione (2)

---

```
int main()
{
    int curMcm, curValue;
    printf("Inserisci il primo valore: ");
    scanf("%d", &curMcm);
    do
    {
        printf("Inserisci un valore: ");
        scanf("%d", &curValue);
        curMCD = mcm(curMcm, curValue);
        printf("mcm corrente: %d\n", curMcm);
    }
    while(curMcm <= 100);
    return 0;
}
```

12

## Esercizio 6

---

### Esercizio 6 - Triangolo di Tartaglia

- Realizzare un programma che, letto in input il massimo livello voluto, mostri a video il contenuto del triangolo di Tartaglia fino a quel livello

13

## Esercizio 6 - Soluzione (1)

---

```
int fattoriale(int n)
{
    int fact = 1, index;
    for(index = n; index > 0; index--)
        fact = fact * index;
    return fact;
}

int binomiale(int n, int k)
{
    return fattoriale(n) / (fattoriale(k) * fattoriale(n-k));
}
```

14

## Esercizio 6 - Soluzione (2)

---

```
int main ()
{
    int Nlivelli, n, k;
    scanf("%d",&Nlivelli);
    for(n = 0; n < Nlivelli; n++)
    {
        for(k = 0; k <= n; k++)
            printf("%d ", binomiale(n, k) );
        printf("\n");
    }
    return 0;
}
```