E5 – Esercizi sui Tipi, variabili ed espressioni in C

Esercizio 1. Scrivere un programma C che chiede all'utente di inserire un numero intero e visualizza all'utente 1 se il numero è pari e 0 se non lo è.

Esercizio 2. Scrivere un programma C che:

- assegna ad una variabile a il più grande valore rappresentabile con il tipo `double` e ad una variabile b il più piccolo valore (positivo) rappresentabile con il tipo `double`;
- assegna ad una variabile c il doppio di a e ad una variabile d la metà di b ;
- stampa c e d .

Che cosa viene stampato?

Esercizio 3. Scrivere un programma C che fa inserire all'utente un numero intero compreso tra 1 e 12 che rappresenta un mese (1=Gennaio, 2=Febbraio, ecc.) e visualizza all'utente 1 se il mese ha trenta giorni, 0 altrimenti.

Esercizio 4. Scrivere un programma C che fa inserire all'utente un numero intero che rappresenta un anno e visualizza all'utente 1 se l'anno è bisestile e 0 se non lo è. Si ricorda che un anno è bisestile se è multiplo di 4. Fanno eccezione gli anni secolari (quelli multipli di 100) che sono bisestili solo se multipli di 400.

Esercizio 5. Dato il seguente programma, dire per quali scelte di valori per a , b e c viene stampato 1.

```
#include <stdio.h>
int main(){
    int a;
    printf("Inserire 0 o 1 per a\n");
    scanf("%d",&a);
    int b;
    printf("Inserire 0 o 1 per b\n");
    scanf("%d",&b);
    int c;
    printf("Inserire 0 o 1 per c\n");
    scanf("%d",&c);
    int d = (a || !b) && (b || !c) && (c || !a);
    printf("d = %d\n",d);
}
```

Soluzioni

Esercizio 1 - svolgimento.

```
#include <stdio.h>

int main(void) {
    printf("Inserisci un intero?\n");
    int n;
    scanf("%d", &n);
    int pari=(n%2==0);
    printf("Numero pari (0=false, 1=true):%d\n", pari);
}
```

Esercizio 2 - svolgimento.

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

int main(void) {

    double a=DBL_MAX;
    double b=DBL_TRUE_MIN;

    double c=2*a;
    double d=b/2;

    printf("c: %g d:%g \n", c, d);
}
```

Verrà stampato

c: 1.#INF d:0

La stringa 1.#INF sta a rappresentare il valore $+\infty$ (la stringa stampata può variare in base al compilatore). Ciò indica che si è avuto un overflow (arrotondato a $+\infty$). Il valore 0 di d indica invece una situazione di underflow (arrotondata a 0).

Esercizio 3 - svolgimento.

```
#include <stdio.h>

int main(void) {

    printf("Inserisci un numero da 1 a 12 che rappresenta un mese\n");
    int mese;
    scanf("%d", &mese);

    int trenta=(mese==4 || mese==6 || mese==9 || mese==11);
    printf("Mese di 30 gg (0=false, 1=true):%d\n", trenta);
}
```

Esercizio 4 - svolgimento.

```
#include <stdio.h>

int main(void) {

    printf("Inserisci un anno\n");
    int anno;
    scanf("%d",&anno);

    int a = ((anno%4)==0); // anno multiplo di 4
    int b = ((anno%100)==0)&&(anno%400!=0); // anno multiplo di 100 ma
                                                // non di 400

    int bisestile = a && !b;

    printf("Anno bisestile (0=false, 1=true):%d\n", bisestile);
}
```

Esercizio 5 - svolgimento.

I valori di a, b e c per cui viene stampato 1 sono a=0, b=0, c=0 e a=1, b=1, c=1. Infatti:

- l'espressione $(a \ || \ !b)$ risulterà vera soltanto nel caso in cui a è vera oppure b è falsa;
 - se a è vera, per rendere vera l'espressione $(c \ || \ !a)$, c deve essere vera, ma allora per rendere vera l'espressione $(b \ || \ !c)$ anche b deve essere vera.
 - se b è falsa, per rendere vera l'espressione $(b \ || \ !c)$, c deve essere falsa, ma allora per rendere vera l'espressione $(c \ || \ !a)$ anche a deve essere falsa.