

Introduzione al C

Corso di Fondamenti di Informatica
Ingegneria Clinica

Esercitazione 6

DIPARTIMENTO DI INFORMATICA
E SISTEMISTICA ANTONIO RUBERTI



SAPIENZA
UNIVERSITÀ DI ROMA

Raffaele Nicolussi

Esercizio 1

Leggere da stdin una sequenza di **0** e **1** terminata da **2** (acquisire i valori uno per volta) e stampare la lunghezza della più lunga sottosequenza di soli **0** presente nella sequenza letta

Esempio: per la sequenza

0 0 1 **0 0 0** 1 1 1 1 0 0 2

la risposta cercata è **3**



Algoritmo Esercizio

Variabili utilizzate (tipo intero):

bit: valore letto, **contatore**: numero di 0 accumulati

lmax: massima lunghezza sottosequenza di 0

- **contatore=0**; **lmax=0** (inizializzazione)
- leggi un numero (valore registrato in **bit**)
- finché **bit** è diverso da 2
 - se **bit** è pari a 0:
 - incrementa **contatore**
 - se **contatore > lmax**: **lmax=contatore**
 - altrimenti
 - contatore=0**
 - leggi un altro numero
- stampa **lmax**

Soluzione

```
int bit, cont = 0, maxlung = 0;
```

```
printf("Inserisci una sequenza di 0 e 1 terminata da 2\n");
```

```
scanf("%d", &bit);
```

```
while ( bit!=2)
```

```
{
```

```
    if ( bit == 0)
```

```
    {
```

```
        cont = cont + 1;
```

```
        if (cont > maxlung)
```

```
            maxlung = cont;
```

```
    }
```

```
    else
```

```
        cont = 0;
```

```
        scanf("%d", &bit);
```

```
}
```

```
printf("La lunghezza della piu\ lunga sottosequenza di 0 e\ %d\n", maxlung);
```




Esercizio 2

Un intero $N > 1$ è detto **primo** se i suoi unici divisori sono **1** e **N**

Scrivere un programma che legge da stdin un intero e determina se è primo

Algoritmo (*inefficiente*): provare se tra i numeri compresi tra **2** e **N-1** c'è un divisore di **N**



Algoritmo(*inefficiente*) Esercizi

Variabili utilizzate (tipo intero):

numero: valore letto, **provadiv**: possibile divisore di **numero**, **trovatodiv**: diventa vero (1) se si trova un divisore di **numero**

- **provadiv=2**; **trovatodiv=0** (inizializzazione)
- leggi valore (registrato in **numero**)
- finché **provadiv** < **numero**
 - se **provadiv** divide **numero**: **trovato=1**
 - **provadiv=provadiv+1**
- se **trovato=1**: **numero** non è primo
altrimenti: **numero** è primo

Soluzione

```
int numero, provadiv = 2, trovatodiv = 0;
```

```
printf ("Inserire un numero intero maggiore di uno: \n");
```

```
scanf ("%d",&numero);
```

```
while (provadiv < numero)
```

```
{
```

```
    if ((numero % provadiv) == 0)
```

```
        trovatodiv = 1;
```

```
    provadiv = provadiv + 1;
```

```
}
```

```
if (trovatodiv==0)
```

```
    printf("Il numero %d e' un numero primo\n", numero);
```

```
else
```

```
    printf("\nIl numero %d non e' un numero primo\n", numero);
```

Esercizio 3

Scrivere un programma che legge da standard input tre numeri interi che rappresentano una data (giorno, mese e anno) e stampa tre interi che rappresentano la data del giorno successivo

- Tenere conto del fatto che i mesi possono essere di 28 (o 29 negli anni **bisestili**), 30 e 31 giorni
- Anno **bisestile**: **divisibile per 400** oppure **divisibile per 4 e non divisibile per 100**. Ad esempio, il 1996 è un anno bisestile, dato che è divisibile per 4 e non è divisibile per 100. L'anno 2000 è un anno bisestile, dato che è divisibile per 400

Algoritmo: Data giorno successivo (1)

1. Leggi giorno, mese, anno
2. Determina numero giorni mese
 - Se mese è 1,3,5,7,8,10,12: giornimese = 31;
 - Se mese è 2: se anno è bisestile,
giornimese = 29, altrimenti giornimese = 28;
 - Se mese è 4, 6, 9, 11: giornimese = 30;
 - Diversamente: mese errato

Algoritmo: Data giorno successivo (2)

3. Determina data giorno successivo

Se mese errato oppure giorno errato:
stampa messaggio d'errore

Altrimenti, se giorno diverso da giornimese:

data giorno successivo = (giorno+1, mese, anno)

Altrimenti, se mese diverso da 12:

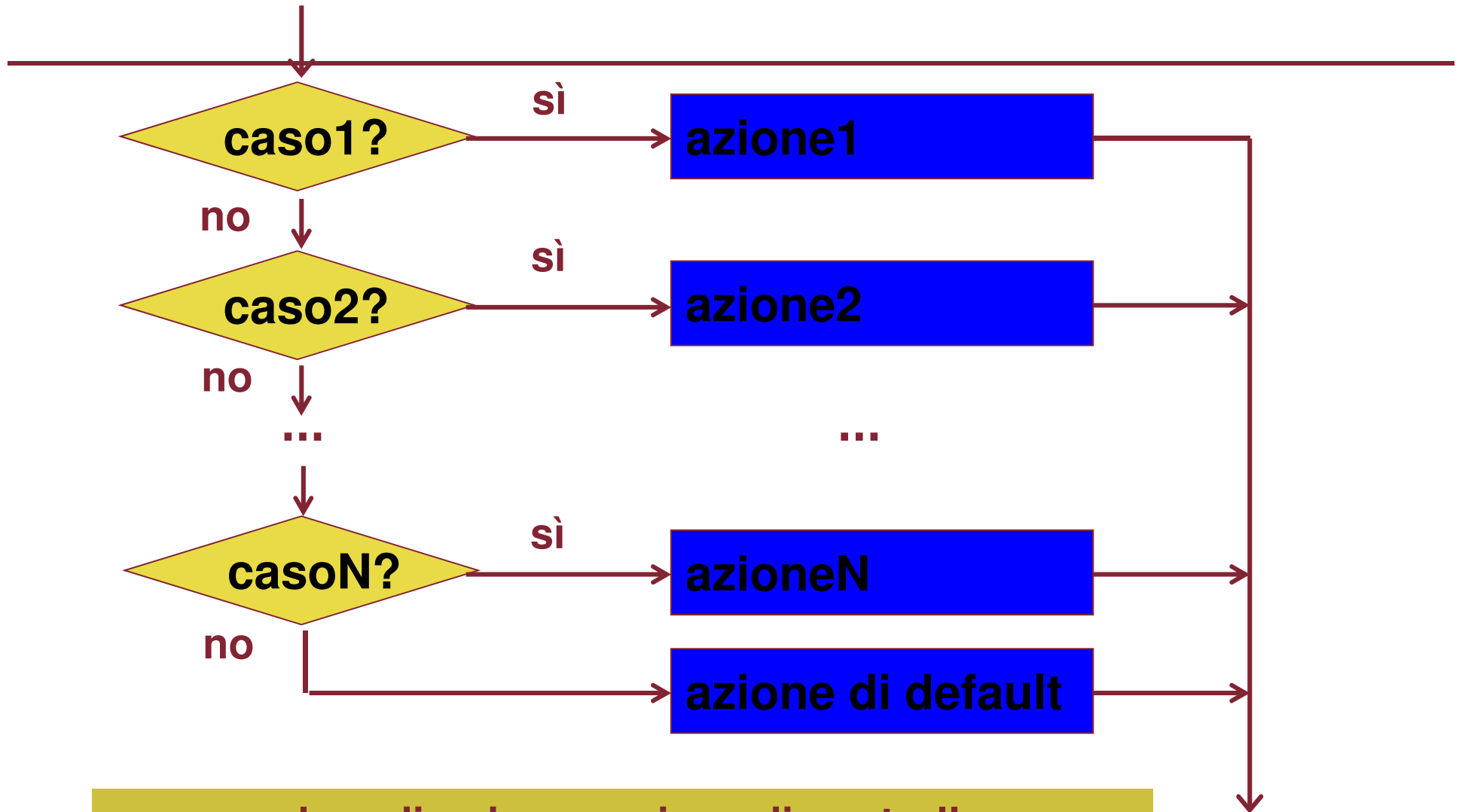
data giorno successivo = (1, mese+1, anno)

Altrimenti:

data giorno successivo = (1, 1, anno+1)

4. Stampa data giorno successivo

Selezione multipla



caso = valore di un'espressione di controllo

Codifica in C (1)

```
int main( ) {
int giorno, mese, anno, giornimese ;
printf( "\nInserire giorno, mese, anno interi\n" ) ;
scanf( "%d%d%d" , &giorno , &mese , &anno ) ;
    switch (mese) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            giornimese = 31; break;
        case 4: case 6: case 9: case 11:
            giornimese = 30; break;
        case 2:
            if (( anno%400==0 ) || (( anno%4==0 ) && ( anno%100!=0
)))
                giornimese = 29 ;
            else
                giornimese = 28 ;
            break;
        default: giornimese = 0; break;
    }
}
```

Codifica in C (2)

```
if ( giornimese == 0 || giorno < 1 || giorno > giornimese )
    printf( "\nData immessa non valida!\n" );
else {
    if ( giorno < giornimese )
        giorno = giorno +1 ;
    else {
        giorno = 1 ;
        if ( mese == 12 ) {
            mese = 1 ;
            anno = anno +1 ;
        }
        else
            mese = mese +1 ;
    }
    printf( "giorno successivo:\n%d,%d,%d\n", giorno, mese, anno );
}
return 0 ;
}
```

Esercizio 4 : Switch

Chiedere all'utente di inserire un intero tra 1 e 12 (estremi inclusi, of course) e stampare il mese dell'anno ad esso corrispondente in formato non numerico

Esempio

Input: 2

Output: Febbraio

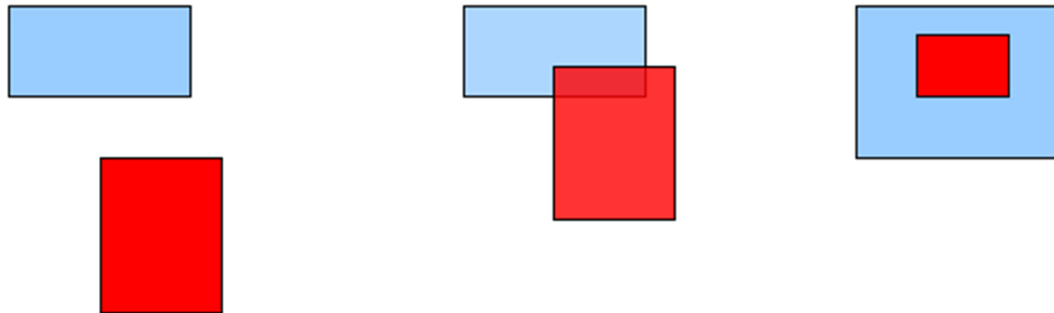
Soluzione Switch

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

int main()
{
    printf("inserisci un intero tra 1 e 12:\n");
    int x;
    scanf("%d", &x);
    switch(x)
    {
        case 1:
            printf("Gennaio\n");
            break;
        case 2:
            printf("Febbraio\n");
            break;
        case 12:
            printf("Dicembre\n");
            break;
        default:
            printf("c'hai provato...\n");
            printf("andra' meglio la prossima volta\n");
            break;
    }
    system("PAUSE");
    return 1;
}
```

Esercizio 5

- Leggere le coordinate di due rettangoli R1 e R2 e stampare
 - 0 se i rettangoli sono disgiunti
 - 1 se si sovrappongono parzialmente
 - 2 se uno contiene l'altro



Soluzione parziale – solo ultimo caso

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

int main()
{
    printf("R1\n");
    printf("inserisci i valori (x, y) per il primo vertice\n"
           " separati da uno spazio:\n");
    int R1_x1, R1_y1;
    scanf("%d %d", &R1_x1, &R1_y1);
    printf("inserisci i valori (x, y) per il secondo vertice\n"
           " separati da uno spazio:\n");
    int R1_x2, R1_y2;
    scanf("%d %d", &R1_x2, &R1_y2);

    int R1_left_bottom_vertex_x, R1_left_bottom_vertex_y,
        R1_right_upper_vertex_x, R1_right_upper_vertex_y;

    if(R1_x1 > R1_x2)
    {
        R1_right_upper_vertex_x = R1_x1;
        R1_left_bottom_vertex_x = R1_x2;
    }
    else
    {
        R1_right_upper_vertex_x = R1_x2;
        R1_left_bottom_vertex_x = R1_x1;
    }
}
```

Soluzione parziale (continua)

```
if(R1_y1 > R1_y2)
{
    R1_right_upper_vertex_y = R1_y1;
    R1_left_bottom_vertex_y = R1_y2;
}
else
{
    R1_right_upper_vertex_y = R1_y2;
    R1_left_bottom_vertex_y = R1_y1;
}

printf("R2\n");
printf("inserisci i valori (x, y) per il primo vertice\n"
      " separati da uno spazio:\n");
int R2_x1, R2_y1;
scanf("%d %d", &R2_x1, &R2_y1);
printf("inserisci i valori (x, y) per il secondo vertice\n"
      " separati da uno spazio:\n");
int R2_x2, R2_y2;
scanf("%d %d", &R2_x2, &R2_y2);

int R2_left_bottom_vertex_x, R2_left_bottom_vertex_y,
    R2_right_upper_vertex_x, R2_right_upper_vertex_y;
```

Soluzione parziale (continua)

```
if(R2_x1 > R2_x2)
{
    R2_right_upper_vertex_x = R2_x1;
    R2_left_bottom_vertex_x = R2_x2;
}
else
{
    R2_right_upper_vertex_x = R2_x2;
    R2_left_bottom_vertex_x = R2_x1;
}

if(R2_y1 > R2_y2)
{
    R2_right_upper_vertex_y = R2_y1;
    R2_left_bottom_vertex_y = R2_y2;
}
else
{
    R2_right_upper_vertex_y = R2_y2;
    R2_left_bottom_vertex_y = R2_y1;
}
```

Soluzione parziale (finisce)

```
if((R2_right_upper_vertex_x < R1_right_upper_vertex_x) &&
    (R2_left_bottom_vertex_x > R1_left_bottom_vertex_x) &&
    (R2_right_upper_vertex_y < R1_right_upper_vertex_y) &&
    (R2_left_bottom_vertex_y > R1_left_bottom_vertex_y))
{
    printf("2 - R2 e' strettamente contenuto in R1\n");
}
else
    printf("R2 NON e' strettamente contenuto in R1\n");

system("PAUSE");
return 1;
}
```



**La stessa cosa
può esser fatta
con R1 ed R2
invertiti**

Esercizio 6 : conta vocali

- Rifare l'esercizio 'conta vocali' usando la switch
- L'esercizio chiedeva di fare inserire una serie di lettere (una per volta) fino all'inserimento del numero 0
- Terminato l'inserimento vanno stampate quante vocali (quante A, quante E, ...) sono state inserite

Esempio switch

```
switch (letter) {  
    case 'A':  
    case 'E':  
    case 'I':  
    case 'O':  
    case 'U':  
        num_vocali++;  
        break;  
    case " ":  
        num_spazi++;  
        break;  
    default:  
        num_altri++;  
        break;  
}
```

**num_vocali++;
equivale all'espressione
num_vocali = num_vocali + 1;**

**Si usa perché è più efficiente
e perché permette di minimizzare
eventuali errori di battitura**

**Il break termina la
valutazione**

Esempio switch (2)

Nel precedente esempio se `letter` e' una vocale ('A','E','I','O','U') viene incrementato il valore della variabile `num_vocali`, se e' uno spazio (" ") si incrementa `num_spazi`, altrimenti (se nessuno dei casi precedenti e' vero) viene eseguita la condizione di default e quindi viene incrementato `num_altri`.