Laboratorio

Uno

1) Espressioni
2) Costrutti Selezione (IF)

Gli argomenti di oggi

- Comandi utili

- Le funzioni: scanf(), printf()
- I tipi, espressioni e scope variabili
- I costrutti di selezione (If)

- Spiegazione esercitazioni e consegne
- Prima Esercitazione

COMANDI UTILI

- Ricordiamo...
  - mkdir nome_directory → per creare una directory
  - emacs programma.c & → avvio di emacs in modalità background e creazione file programma.c
  - gcc programma.c → per compilare il programma
  - ./a.out → per eserguirlo dopo la compilazione
  - gcc programma.c –o programma → compilare con nome all'eseguibile
  - ./programma → per eseguirlo

PRINTF.

*int printf (const char * format, ...);*

- **format**
  - string that contains the text to be written to stdout. It can optionally contain embedded format tags that are substituted by the values specified in subsequent argument(s) and formatted as requested.
  - the number of arguments following the *format* parameters should at least be as much as the number of format tags.
- **additional arguments**
  - depending on the *format* string, the function may expect a sequence of additional arguments, each containing one value to be inserted instead of each %-tag specified in the *format* parameter, if any. There should be the same number of these arguments as the number of %-tags that expect a value.

fonte

- http://www.cplusplus.com/reference/clibrary/cstdio/printf/

PRINTF..

- Esempi
  - `printf ("Characters: %c %c \n", 'a', 65);`
    - `Characters: a A`
  - `printf ("Decimals: %d %ld\n", 1977, 650000L);`
    - `Decimals: 1977 650000`

I Tipi.

```
#include <stdio.h>

void main(){
   // dichiarazione e inizializzazione
    int sono_int = 0;
    float sono_float = 0.0;

    // assegnamento nuovo valore
    sono_int = 3;
    sono_float = 3.5;

    // stampa
    printf("sono_int=%d  \t  sono_float=%f  \n",  sono_int,
      sono_float);
}
```

```
sono_int = 3  sono_float = 3.5
```

I Tipi..

- Operazioni fondamentali (più usate)
  - somma (+); differenze (-); prodotto (*); rapporto (/)
  - operatori di incremento (++) e decremento (--)
- Risultati delle operazioni

	Operazione	Risultato	Esempio	Operazione	Risultato	Esempio
Somma	int + int	int	5 + 3 = 8	float + float	float	5.4 + 3.3 = 8.7
Differenza	Int – int	Int	4 – 9 = 5	float – float	float	4.2 – 2.1 = 2.1
Prodotto	Int * int	Int	2 * 4 = 8	float * float	float	2.1 * 4 = 8.4
Divisione	Int / int	int	5/2 = 2	float /float	float	5.0/2.0 = 2.5

I Tipi

```
#include <stdio.h>

void main(){
    // dichiarazione e inizializzazione delle due variabili
    int sono_int = 0;
    float sono_float = 0.0;

    // operazione divisione tra int e float
    sono_int = 7/2;
    sono_float = 7/2;
    printf("7/2 ->\tsono_int=%d \n\tsono_float=%f \n", sono_int, sono_float);

    sono_int = 7.0/2;
    sono_float = 7.0/2;
    printf("7.0/2 ->sono_int=%d \n\tsono_float=%f \n", sono_int, sono_float);
    printf("invertendo i percentuali nella stampa:\nsono_int=%f \nsono_float=%d \n",
    sono_int, sono_float);
}
```

```
7/2 ->   sono_int=3        sono_float=3.000000
7.0/2 ->sono_int=3         sono_float=3.500000
```

OPERATORI ++ E --.

```
#include <stdio.h>

void main(){
   // dichiarazione e inizializzazione delle due variabili
   int i = 0;
   int j = 0;
   int k = 0;

   printf("Valori iniziali:\n");
   printf("i=%d \t j=%d \t k=%d \n", i, j, k);
   // i=0   j=0   k=0

   i++;
   ++j;
   printf("i++ e ++j\n");
   printf("i=%d \t j=%d \t k=%d \n", i, j, k);
   // i=1   j=1   k=0
```

OPERATORI ++ E --..

```
k = ++i;
printf("k = ++i\n");
printf("i=%d \t j=%d \t k=%d \n", i, j, k);
// i=2  j=1  k=2

k = j++;
printf("k = j++\n");
printf("i=%d \t j=%d \t k=%d \n", i, j, k);
// i=2  j=2  k=1

k -= i++;
printf("k -= i++\n");
printf("i=%d \t j=%d \t k=%d \n", i, j, k);
  // i=3  j=2  k=-1
```

OPERATORI ++ E --...

```
k *= ++j;
printf("k *= ++j\n");
printf("i=%d \t j=%d \t k=%d \n", i, j, k);
// i=3  j=3  k=-3

k *= --j;
printf("k *= --j\n");
printf("i=%d \t j=%d \t k=%d \n", i, j, k);
// i=3  j=2  k=-6

printf("i=%d \t j=%d \t k=%d \n", i++, --j,
k);
// i=3  j=1  k=-6
```

SCOPE VARIABILI

- I blocchi di istruzioni in C vengono definiti mediante le parentesi graffe.
- I blocchi possono essere annidati e se ne possono definire un numero arbitrario
- Le variabili sono visibili solo dal momento della dichiarazione fino al termine del blocco dove sono dichiarate e quindi anche in tutti i blocchi più interni
- Quindi via via che il programma procede solo alcune delle variabili dichiarate nell'intero programma sono visibili e utilizzabili.
- Attenzione: una variabile dichiarata dentro un blocco con lo stesso nome di una variabile dichiarata in un blocco più esterno, la nasconde!

ESERCIZIO.

```
#include <stdio.h>
void main(){
   int x = 1;
   int y = 2;
   int z = 3;

   printf("x = %d y = %d z = %d\n",x,y,z);
   // x = 1 y = 2 z = 3

   {
       x = x + y;
       z = 5;
       printf("x = %d y = %d z = %d\n",x,y,z);
       // x = 3 y = 2 z = 5
   }
   printf("x = %d y = %d z = %d\n",x,y,z);
   // x = 3 y = 2 z = 5
```

ESERCIZIO..

```
{
    int x = 10;
    x = x + y;
    z = x;
    printf("x = %d y = %d z = %d\n",x,y,z);
    // x = 12 y = 2 z = 12
}

printf("x = %d y = %d z = %d\n",x,y,z);
// x = 3 y = 2 z = 12
}
```

Costrutto If.

- *permette l'esecuzione condizionale di un'istruzione (o blocco di istruzioni) in base al valore di verità di una condizione specificata (espressione logica)*

```
if (condizione) {
/* blocco eseguito se la condizione è
  vera */
} else {
/* blocco eseguito se la condizione è
  falsa */
}
```

ESERCIZIO

Scrivere un programma che *richiede da input sesso (M o F) ed età (int) e stampare a video una delle due frasi:*

- *sei un maschietto e hai x anni*
- *sei una femminuccia e hai x anni*

SOLUZIONE

```
#include <stdio.h>

void main(){
    char sex;
    int eta;
    printf("Inserire il proprio sesso:\n M sta per
      maschio:\n F sta per femmina:\n");
    scanf("%c", &sex);
    printf("Inserire l'eta\n");
    scanf("%d", &eta);

    if((sex=='m') || (sex== 'M') )
      printf("Sei un maschio e hai %d anni", eta);
    else
      printf("Sei una femmina e hai %d anni",eta);
}
```

SOLUZIONE MIGLIORATA

```
#include <stdio.h>

void main(){
    char sex;
    int eta;
    printf("Inserire il proprio sesso:\n M sta per
      maschio:\n F sta per femmina:\n");
    scanf("%c", &sex);
    printf("Inserire l'eta\n");
    scanf("%d", &eta);

    if((sex=='m') || (sex== 'M') )
      printf("Sei un maschio e hai %d anni", eta);
    else if((sex=='f') || (sex== 'F') )
      printf("Sei una femmina e hai %d anni",eta);
    else printf("Errore"); // gestione errori
}
```

ESERCIZIO

Scrivere un programma che:

- chiede all'utente il sesso e l'età:
  - se persona con anni <21 allora non può entrare in discoteca
  - se maschio con anni >=21 allora può entrare e paga 10 $
  - se femminuccia con anni >=21 entra gratis
- Il programma deve stampare a video la posizione dell'utente!

SOLUZIONE

```
#include <stdio.h>

void main(){
    char sex;
    int eta;

    printf ("sei maschio o femmina?");
    scanf("%c",&sex);
    printf("quanti anni hai?");
    scanf("%d",&eta);

    if (eta<21) printf ("non puoi entrare");
    else { if ((sex=='m')||(sex=='M'))
                printf("paghi 10$");
           else
                printf("entri gratis");
    // per semplicita' omettiamo il controllo sull'input...
    } //end blocco else dell'if principale.
} // end main
```

SOLUZIONE MIGLIORATA

```
#include <stdio.h>
void main(){
    char sex;
    int eta;

    printf ("sei maschio o femmina?");
    scanf("%c",&sex);
    printf("quanti anni hai?");
    scanf("%d",&eta);

    if (eta<21) printf ("non puoi entrare");
    else { if ((sex=='m')||(sex=='M'))
                printf("paghi 10$");
           else if ((sex=='f')||(sex=='F'))
                printf("entri gratis");
           else printf("Errore \n");
    // per semplicita' omettiamo il controllo sull'input...
    } //end blocco else dell'if principale.
} // end main
```

RAND.

- La funzione rand() genera numeri pseudocasuali compresi tra 0 e RAND_MAX (numero che dipende dal sistema in cui lavoriamo)

- Si deve includere la libreria:

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

RAND..

- Programma che simula il lancio di un dado

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

void main () {
   //genera un numero casuale fra 0 e 5
    int esito = rand() % 6;

    esito++; //otteniamo un valore fra 1 e 6

    //stampa del risultato
    printf("Lancio del dado = %d\n",esito);
    printf(" Il valore di RAND_MAX e' %d\n",RAND_MAX);
}
```

RAND…

- Proviamo ad eseguire il programma più volte…

  si ottiene sempre lo stesso risultato!!!

- **NON è un BACO del sistema, ma semplicemente rand() genera numeri PSEUDO-CASUALI, cioè numeri deterministici che "sembrano" casuali.**

- Una prova:
  - simulare il lancio di due dadi all'interno dello stesso programma
  - otterremo risultati diversi

RAND....

- Per fare in modo (se lo vogliamo) che ogni esecuzione del programma generi un diverso valore del lancio del dado, dobbiamo variare il seme usato per **inizializzare il generatore di** numeri pseudo-casuali (randomizzazione).
- La randomizzazione viene fatta tramite la funzione srand(), che richiede un parametro di input di tipo unsigned (intero senza segno).
- Per generare il seme, tipicamente viene sfruttata la funzione time() della libreria time.h che, invocata con parametro NULL restituisce un valore unsigned che rappresenta l'ora corrente del giorno espressa in secondi

RAND.....

- Programma lancio del dado nella versione randomizzata

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

void main () {
    int esito;

    //inizializza il generatore
    srand(time(NULL));

    esito = rand() % 6;
    esito++; //otteniamo un valore fra 1 e 6
    //stampa del risultato
    printf("Lancio del dado = %d\n",esito);
}
```

Esercizio

- Scrivere un programma che stampa a video la somma e il prodotto di due numeri (casuali e compresi tra 0 e 9 compresi) e chiede all'utente di indovinare e inserire i due numeri segnalando se sono giusti oppure no

SOLUZIONE

```
#include <stdio.h>
#include<time.h>
#include <stdlib.h> //per random
    void main() {
    int n1 = 0, n2 = 0;
    srand (time (NULL));
    n1= rand()%10; //in [0,..,9]
    n2= rand()%10;
    int ins1, ins2;
    int somma = n1+n2, prodotto = n1*n2;
    printf("Somma=%d,  prodotto=%d:  che  numeri  sono?\n",  somma,
       prodotto);
    scanf("%d", &ins1);
    scanf("%d", &ins2);
    if ((ins1 == n1 && ins2 == n2) || (ins1 == n2 && ins2 == n1)) {
       printf("Indovinato!\n");
    }
    else{
       printf("Sbagliato!\n");
       printf("La risposta corretta e': %d e %d\n",n1, n2);
    }
}
```

PRIMA CONSEGNA

Regole Consegne (1)

- Le consegne sono facoltative
  - ciascuno può decidere se consegnare oppure no
- Dopo ogni consegna vi sarà una selezione casuale degli iscritti al corso
  - gli elaborati dei selezionati verranno valutati
  - agli elaborati di chi ha consegnato ma non è stato estratto verrà assegnato un voto pari alla media dei voti degli estratti
  - gli studenti che consegneranno un elaborato vuoto o che non si attiene minimamente alla consegna verranno eliminati dalle estrazioni
- I voti saranno pubblicati al termine del corso

Regole Consegne (2)

- Per consegnare bisogna creare una directory apposita, includere in essa i file da consegnare
- I file si dovranno chiamare esercizio_uno, esercizio_due,…
- Da laboratorio oppure anche collegandosi in SSH (https://support.math.unipd.it/?q=node/29) digitare da dentro la cartella creata (che contiene solo i file da consegnare) il comando:

consegna consegna1

Prima Consegna

- La prima consegna dovrà essere eseguita entro il giorno 1 dicembre 2011
- Le consegne effettuate successivamente non verranno considerate...

Esercizio 1

- Scrivere un programma “esercizio_1.c” che simuli il gioco della morra cinese.
- Il programma dovrà prendere in ingresso due nomi con cui identificherà i due giocatori. I nomi verranno memorizzati in variabili di tipo char.
- Al primo giocatore è richiesto di inserire una parola tra forbice (‘f’), carta (‘c’) o pietra (‘p’)
- Al secondo giocatore viene chiesta la medesima cosa.
- Il programma dovrà restituire in output il nome del vincitore.
- Se la parola in ingresso non è una delle tre previste il programma dovrà avvisare che è stato commesso un errore.

ESERCIZIO 2

- Scrivere un programma “esercizio_2.c” che data la velocità del vento (v) in nodi (definita come numero intero letto da tastiera), ne classifichi la forza secondo la seguente scala:
  - $v < 1$: calmo
  - $1 \leq v \leq 3$: bava di vento
  - $4 \leq v \leq 27$: brezza
  - $28 \leq v \leq 47$: burrasca
  - $48 \leq v \leq 63$: tempesta
  - $v > 63$: uragano
- e stampi a video velocità e forza.