Dipartimento di Informatica e Sistemistica
Antonio Ruberti

“Sapienza” Università di Roma

Esercitazione 3

Corso di Fondamenti di Informatica

Laurea in Ingegneria Informatica

(Canale di Ingegneria delle Reti e dei Sistemi Informatici - Polo di Rieti)

Anno Accademico 2007/2008

Tutor: Ing. Diego Rughetti

Problemi con Windows Vista:soluzione

Nel menù Strumenti => Opzioni di Compilazione:

- aggiungere C:\Dev-Cpp\libexec\gcc\mingw32\3.4.2 ai File Binari nel tab Cartelle;

- aggiungere il prefisso C:\dev-cpp\bin\ ad ogni percorso nel tab Programmi;

Esercizio 1

Scrivere un programma che prende in ingresso una serie di sequenze numeriche e ne stampa a video la media. Ogni sequenza numerica in ingresso termina quando viene immesso il numero 0. I numeri immessi possono essere sia positivi che negativi. Si vuole stampare a video la media di ogni singola serie numerica e infine la media di tutte le medie.
La serie di sequenze numeriche termina quando vengono immessi due 0 di seguito (il primo 0 termina l'ultima sequenza, il secondo 0 termina la serie di sequenze)

Soluzione

```
#include <stdio.h>
int main(){
        int media, int n, int k, int j;
        float mediaMedie;
        media = mediaMedie = n = k = j = 0;
        printf("n = ");
        scanf("%d",&n);
        do{
                while (n){
                        media = media + n;
                        k++;
                        printf("n = ");
                        scanf("%d",&n);
                }
                if(k >0)
                        printf(“fine serie, media = %12.3f\n”, (float) media/k);
                        mediaMedie = mediaMedie + media/k;
                        j++;
                        k = 0;
                else
                        printf(“immessa serie vuota”);
                printf("n = ");
                scanf("%d",&n);
        }while(n);
        if(j>0)
                printf(“media delle medie = %12.6f\n”, mediaMedie/j);
        else
                printf(“immesse solo sequenze vuote”);
}
```

Esercizio 2

Scrivere un programma che prende in ingresso una serie di sequenze numeriche e stampa a video il maggiore ed il minore. Ogni sequenza numerica in ingresso termina quando viene immesso il numero 0. I numeri immessi possono essere solo positivi. Si vuole stampare a video il maggiore ed il minore di ogni singola serie numerica e infine il maggiore ed il minore generali.
La serie di sequenze numeriche termina quando vengono immessi due 0 di seguito (il primo 0 termina l'ultima sequenza, il secondo 0 termina la serie di sequenze)

Soluzione

```
#include <stdio.h>
int main(){
    int max, int min, int gMax, int gMin;
    int i = 0;
    do{
        printf("n = ");
        scanf("%d",&n);
        if(n == 0) i++;
        if(i == 2) {
            printf(“sequenze tutte vuote”);
            return 1;
        }
    } while(n <= 0)
    min = max = gMax = gMin = n;
    do{
        while (n){
            if(n>max)
                max = n;
            else if(n<min)
                min = n;
            printf("n = ");
            scanf("%d",&n);
        }
        printf(“Max sequenza = %d\n”, max);
        printf(“Min sequenza = %d\n”, min);
        if (max>gMax)
            gMax = max;
        if(min<gMin)
            gMin = min;
        printf("n = ");
        scanf("%d",&n);
        if(n>0){
            max = min = n;
        }
    }while(n);
    printf(“Max assoluto = %d\n”, gMax);
    printf(“Min assoluto = %d\n”, gMin);
}
```

Completare la tabella, effettuando le relative conversioni di base:

Decimale	Binario	Esadecimale
0	00000000	00
55	**00110111**	**37**
136		
243		
	01010010	
	10101100	
	11100111	
		A7
		3E
		BC

Conversione Decimale-Binario:

55 mod 2 =1 = a0

(55 div 2) mod 2 = 27 mod 2 = 1 = a1

((55 div 2) div 2) mod 2= (27 div 2) mod 2=13 mod 2=1= a2

(13 div 2) mod 2= 6 mod 2 = 0 =a3

(6 div 2) mod 2 = 3 mod 2 = 1 =a4

(3 div 2) mod 2 = 1 mod 2 = 1 = a5

Dunque:

$55_{10}=110111_2$

Conversione Binario-Esadecimale:

1) Raggruppo i bit a blocchi di 4:
   11 0111
2) Converto ciascun blocco in base 16
   $11_2=3_{16}$ $0111_2=7_{16}$

Dunque: $110111_2 = 37_{16}$

Completare la tabella, effettuando le relative conversioni di base:

Decimale	Binario	Esadecimale
0	00000000	00
55	**00110111**	**37**
136		
243		
82	01010010	**52**
	10101100	
	11100111	
		A7
		3E
		BC

Conversione Binario-Decimale:

01010010 $=0*2^0+1*2^1+0*2^2+0*2^3+1*2^4+0*2^5+1*2^6+0*2^7=$
$=2+16+64=82_{10}$

Conversione Binario-Esadecimale:

1) Raggruppo i bit a blocchi di 4:
   0101 0010
2) Converto ciascun blocco in base 16
   $0101_2=5_{16}$ $0010_2=2_{16}$
   $=>52_{16}$

Completare la tabella, effettuando le relative conversioni di base:

Decimale	Binario	Esadecimale
0	00000000	00
55	**00110111**	**37**
136		
243		
82	01010010	**52**
	10101100	
	11100111	
167	**10100111**	A7
62		3E
		BC

Conversione Esadecimale-Decimale:

A7 $=10*16^1+7*16^0=10*16+7=167$

3E $= 3*16^1+E*16^1=48+14=62$

Conversione Esadecimale-Binario:

1) Traduco ciascuna cifra esadecimale in un blocco di 4 bits :

$A_{16} = 10_{10}= 2^3+2^1=1010_2$

$7_{16} = 7_{10}= 2^2+2^1+2^0=0111_2$

2) Il binario corrispondente si ottiene sostituendo ciascuna cifra esadecimale con il relativo blocco di 4 bit:

$A7_{16}=10100111_2$

Somma tra Numeri Binari Interi Positivi

Esempio:

0 0 0 0 1 1 1 0 *riporti*

0 0 1 0 1 1 1 0 +

0 1 0 0 0 1 1 1 =

---

0 1 1 1 0 1 0 1

Un trabocco indica un overflow

1 0 0 0 1 1 1 0 *riporti*

1 0 1 0 1 1 1 0 +

1 1 0 0 0 1 1 1 =

---

1 1 1 1 0 1 0 1

Sottrazione tra Numeri Binari Interi Positivi

Esempio:

0 1 0 0 0 1 1 1 *prestiti*

1 0 1 0 1 1 1 0 -

0 1 0 0 0 1 1 1 =

---

0 1 1 0 0 1 1 1

Un trabocco indica un underflow.

1 1 0 1 1 1 1 1 *prestiti*

1 0 1 0 0 1 1 0 -

1 1 0 0 0 1 1 1 =

---

1 1 0 1 1 1 1 1

Moltiplicazione tra Numeri Binari Interi Positivi: Potenze di 2

```
  00110 x
     10 =
 -------
  00000
 00110
-------
 001100
```

```
  00110 x
    100 =
 -------
  00000
 00000
00110
-------
0011000
```

Somma tra Numeri Binari in Complemento a 2

$0010_2+1101_2=1111_2$

riporto 0000

0010+

1101=

1111

$0110_2+0111_2=$?

riporto 0110

0110

0111

1101

Somma tra Numeri Binari in Complemento a 2

1111_2+1000_2= ?

riporto 1000

1111+

1000=

0111

$0110_2+1111_2=0111$

riporto 1110

0110

1111

0101

Esercizio 3

Scrivere un programma C che prende in ingresso una data (giorno, mese ed anno) e stampa la data successiva. Modularizzare il programma suddividendolo nelle seguenti funzioni:

- -leggiGiorno()
- -leggiMese()
- -leggiAnno()
- -giorniMese()
- -calcolaDataSuccessiva()

Soluzione (1)

```
#include <stdio.h>

int g,m,a;

int giornimese(int, int);
void valcoladatasuccessiva();
int leggigiono();
int leggimese();
int leggianno();

Int main(){
        g = leggigiorno();
        m = leggimese();
        a = leggianno();
        printf("Data: %d/%d/%d\n",g,m,a);
        calcoladatasuccessiva();
        printf("Data successiva: %d/%d/%d\n",g,m,a);
}
```

Soluzione (2)

```
int leggigiorno(){
		int m;
		do{
			printf("giorno:"); scanf("%d",&m);
		}while (m < 0 || m > 31);
	return m;
}

int leggimese(){
		int m;
		do{
			printf("mese:"); scanf("%d",&m);
		}while (m < 0 || m > 12);
	return m;
}
int leggianno(){
		int m;
		do{
			printf("anno:"); scanf("%d",&m);
		}while (m < 0);
	return m;
}
```

Soluzione (3)

```
void calcoladatasuccessiva(){
		int giorni;
		giorni = giornimese(m,a);
		g = g+1;
		if (g > giorni){
				g = 1;
				m++;
				if (m > 12){
						m = 1;
						++a;
				}
		}
}

int giornimese(int m, int a){
		if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12)
				return 31;
		else if (m == 2)
				if ( (a % 4 == 0) && (a % 100 != 0) || (a % 400 == 0) )
						return 29;
				else
						return 28;
		else
				return 30;
}
```

Esercizio 4

Definire una funzione che prenda in ingresso un numero intero e restituisca il fattoriale di tale numero. Fornire anche un esempio di main per testare la funzione.

Soluzione

```
#include <stdio.h>

long int fact(int n);
void main(){
        int m;
        long int fattoriale;
        printf(“inserire un numero:");
        scanf("%d",&m);
        fattoriale = fact(m);
        printf(“fattoriale = %ld”, fattoriale)
}

long int fact(int n){
        long int fact;
        int i;
        fact = 1;
        for (i = 1; i <= n; i++)
                fact *= i;
        return fact;
}
```

Esercizio 5

Realizzare una funzione che, dato un numero intero, verifichi se esso è un numero primo. Utilizzando poi la funzione appena realizzata, scrivere un programma che, dato un numero intero positivo, visualizzi a video tutti i numeri primi minori o uguali di quello dato.

Soluzione (1)

```
int IsPrimeNumber(int n) {

// Sfrutta direttamente la definizione di numero primo:
// scarsamente efficiente
// Controllo se il numero e' minore o uguale di zero:
// numero non primo, uscita immediata.

        if (n <= 0)
                return 0;

// provo a dividere n per tutti i numeri da 2 a n-1;
// se almeno una divisione ha resto nullo (divisibilità)
// il numero non e' primo, uscita immediata.

        for (int i = 2; i < n; i++)
                if (n % i == 0) return 0;

        return 1;

};
```

```
int IsPrimeNumber2(int n) {

// Versione piu' efficiente:
// - rileva se n e' pari e diverso da 2: in quel caso non è primo;
// - poi prova a dividere solo per i numeri dispari.
// Inoltre divide solo fino ad n/2 - 1: le successive divisioni
// sarebbero superflue.
// Controllo se il numero e' minore o uguale di zero: numero non
// primo, uscita immediata.

        if (n <= 0)
                return 0;

        if (n <= 2)
                return 1;

//Controllo se n è pari. In questo caso il numero non è primo.

        if (n % 2 == 0) return 0;

        for (int i = 3; i < n/2; i = i + 2)
                if (n % i == 0) return0;

        return 1;

};
```

Soluzione (2)

```
#include <stdio.h>

int IsPrimeNumber(int n);
int IsPrimeNumber2(int n);

int main() {

		int LastNum;

		do {
			printf( "Calcolo dei numeri primi fino a: “);
			scanf(“%d”, &LastNum);

		} while (LastNum <= 0);

		for (int i = 1; i <= LastNum; i++){
			if (IsPrimeNumber2(i))
				printf(“%d; ”, i);
		}
		printf(“fine\n”);
		system("PAUSE");
		return 1;
};
```