



**Dipartimento di Informatica e Sistemistica
Antonio Ruberti**

“Sapienza” Università di Roma

Esercitazione 5

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

Ricorsione

Esercizio 1

Scrivere una funzione C che prenda in ingresso un numero decimale n ed un numero decimale b e stampi a video la rappresentazione di n in base b . Scrivere anche un main per testare la funzione.

Soluzione

```
void d2b(int n,int b){
    if (n >= b){
        d2b(n/b,b);
        printf("%d",n%b);
    }
    else
        printf("%d",n);
}
```

Esercizio 2

Progettare e codificare un programma in C che calcoli il quoziente della divisione tra interi. Ovvero scrivere una funzione RICORSIVA che prende in ingresso due interi x , y e restituisce il quoziente della divisione x/y e un main per provare il suo funzionamento.

Soluzione

```
#include <stdio.h>

int div(int x, int y){
    if (x>=y)
        return 1+div(x-y,y);
    else
        return 0;
}

main() {
    int x,y,ris;
    do{
        printf("Inserire dividendo: ");
        scanf("%d",&x);
    }while (x<0);
    do{
        printf("Inserire divisore: ");
        scanf("%d",&y);
    }while (y<1);
    ris = div(x,y);
    printf("%d/%d = %d",x,y,ris);
}
```

Algoritmo ricorsivo per il calcolo del quoziente:

$$x/y = (x - y + y)/y = 1 + (x - y)/y.$$

Esercizio 3

Scrivere una funzione C che calcola $\sin(x)$ utilizzando lo sviluppo di Taylor fino al termine n-esimo. Realizzare utilizzando la ricorsione sia la funzione per il calcolo del fattoriale, sia la funzione che calcola $\sin(x)$. Fornire anche un main per testare la funzione.

Suggerimento:

$$\sin(x) \approx \sum_{k=0}^n (-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

Soluzione

```
#include <stdio.h>
#include <math.h>
#define PI 3.14159
```

```
int fatt(int n){
    if (n>1)
        return n*fatt(n-1);
    else
        return 1;
}
```

```
double sin_taylor(double x, int n){
    if (n>=0)
        return ( pow(-1,n)*pow(x,2*n+1)/fatt(2*n+1) + sin_taylor(x,n-1) );
    else
        return 0;
}
```


Soluzione (2)

```
main(){
    double x,ris;
    int n;
    do{
        printf("Inserire angolo in radianti: ");
        scanf("%lf",&x);
    }while (x<0 || x>=2*PI);

    printf("Inserire grado del polinomio: ");
    scanf("%d",&n);
    ris = sin_taylor(x,n);
    printf("sin_taylor(%lf,%d) = %lf\n",x,n,ris);
    printf("sin(%lf) = %lf",x,sin(x));
}
```

Array

Esercizio 4

Scrivere una funzione che prenda in ingresso tre parametri: un array e due numeri interi n e x . La funzione deve verificare se il numero x è presente all'interno dell'array. Se c'è allora la funzione deve restituire l'indice del numero, altrimenti -1

Soluzione

```
int search1(int A[],int n, int x){
    int i;
    for (i=0; i < n; i++){
        if (A[i] == x)
            return i;
    }
    return -1;
}
```

```
int search2(int A[],int n, int x){
    int i = 0, found = -1;
    while (i < n && found == -1){
        if (A[i] == x)
            found = i;
        else i++;
    }
    return found;
}
```

Esercizio 5

Scrivere un programma c che prende in input da tastiera un numero decimale intero positivo, lo converte in binario e stampa a video il risultato.

Attenzione, la stampa a video deve esser fatta posizionando correttamente le cifre del risultato, ovvero partendo dall'MSB posizionato più a sinistra fino ad arrivare all'LSB nella posizione più a destra.

Soluzione

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

```
int main () {  
    int i, num;  
    int numero_binario [16]; // per interi a 16 bit  
    int cifre_inserite; // contatore delle cifre binarie effettivamente inserite  
    cifre_inserite = 0;  
    do {  
        printf( "inserisci un numero maggiore o uguale a 0\n");  
        scanf("%d",&num);  
    }while(num<0);  
    printf ("la conversione in binario e': ");  
    do{  
        numero_binario[cifre_inserite] = num%2;  
        cifre_inserite++;  
        num=num/2;  
    }while (num!=0);  
    for(i=cifre_inserite-1; i>=0; i--)  
        printf("%d", numero_binario[i]);  
}
```

Esercizio 6

Scrivere un programma c che prende in ingresso da tastiera una serie di 20 numeri interi. I numeri immessi devono essere salvati all'interno di un array.

Una volta terminata l'immissione dei numeri trovare all'interno dell'array il massimo ed il minimo e stamparli a video. Dopo aver fatto ciò, ordinare l'array in ordine decrescente e successivamente stamparlo a video.

(Gli algoritmi per l'ordinamento, il massimo ed il minimo sono già stati visti a lezione, comporli in un solo main)

Soluzione

```
int main(){
    int myArray[20];
    int temp1, temp2, i, j;
    int max, min;
    ..... immissione da tastiera e salvataggio in un array
    max = myArray[0];
    min = myArray[0];
    for (i = 1; i<20;i++){
        if(max < myArray[i])
            max = myArray[i];
        if(min > myArray[i])
            min = myArray[i];
    }
    printf("max = %d, min = %d", max, min);
    //ordinamento e stampa dell'array
    for (i = 0; i<20;i++){
        for(j = (i+1); j<20; j++){
            if (myArray[i]<myArray[j]){
                temp1 = myArray[i];
                temp2 = myArray[j];
                myArray[i] = temp2;
                myArray[j] = temp1;
            }
        }
        printf("%d ", myArray[i]);
    }
}
```