Fondamenti di Informatica
Esercitazione 11

29 novembre 2022

Allocazione Dinamica

11.1 Compilate e eseguite il seguente codice:

```
1  // Introduzione a  malloc() e free()
2
3  #include <stdlib.h>     // Libreria contenente le funzioni
4  #include <stdio.h>
5
6
7  main (int argc)
8  {
9  char *pString;          // Creiamo un pointer a un char
10
11
12 pString = malloc(10);       // Prendiamo 10 byte di memoria
13 pString[0] = 'H';
14 pString[1] = 'e';
15 pString[2] = 'l';
16 pString[3] = 'l';
17 pString[4] = 'o';
18 pString[5] = '\0';      // Terminatore di stringa
19
20 printf("%s\n", pString);
21
22 *(pString+1) = 'a';     // Aritmetica dei puntatori
23                 //
24
25 printf("%s\n", pString);    //
26
27 free(pString);          // Liberiamo la memoria
28
29
30
```

```
31 }
32
33 //============================================================================
```

Cosa fa il codice?
Cosa succede se non allochiamo abbastanza memoria per la stringa?
Cosa succede se non de-allochiamo free?

11.2 Scrivere un sottoprogramma che crea un array di interi dimensione N, allocando la memoria dinamicamente.
Scrivere un sottoprogramma che in input un array di interi e la sua dimensione N e trova il massimo elemento usando l'aritmetica dei pointer.
Scrivere un sottoprogramma che prende in input un array e libera la sua memoria.
Scrivere un programma che chiede all'utente di inserire un array di interi e stampa a video il suo elemento massimo, usando i sottoprogrammi precedenti.

11.3 Scrivere un sottoprogramma che riceve come parametro un intero positivo N, alloca dinamicamente e restituisce una matrice quadrata NxN di valori interi. Il sottoprogramma restituisce NULL nel caso di errore di allocazione (deallocando l'eventuale memoria già allocata).
Scrivere un sottoprogramma che riceve come parametro una matrice bidimensionale di valori interi allocata dinamicamente ed il suo numero di righe, e ne libera la memoria.
Scrivere un programma che chiede all'utente la dimensione di una matrice quadrata m e poi i dati per popolarla; non sapendo a priori le dimensioni della matrice m, il programma allocherà la matrice mediante il sottoprogramma sopra definito. Il programma cerca nella matrice m la sottomatrice di dimensione massima che parte da m[0][0] e la cui somma dei valori è pari a zero e, se esiste, la salva in una seconda matrice allocata anch'essa dinamicamente. In seguito, il programma stampa a video la nuova matrice calcolata (se esiste), libera la memoria e termina. Gestire opportunamente gli eventuali errori nell'allocazione della memoria.
Esempio: se l'utente inserisce la matrice 3x3:
1 2 3
-1 -2 3
3 3 3
La sottomatrice a somma nulla di dimensione massima che parte da m[0][0] è:
1 2
-1 -2

11.4 Scrivere un sottoprogramma che riceve come parametro un intero positivo N, alloca dinamicamente e restituisce una matrice quadrata NxN di valori float. Il sottoprogramma restituisce NULL nel caso di errore di allocazione (deallocando l'eventuale memoria già allocata).

Scrivere un sottoprogramma che riceve come parametro una matrice bidimensionale di valori float allocata dinamicamente ed il suo numero di righe, e ne libera la memoria.
Si consideri un file di testo serie.txt che contiene una sequenza di numeri. Il primo valore, un intero n, indica quanti numeri float sono di seguito contenuti nel file; si assuma che il file sia ben formato (cioè il numero di valori contenuti è n+1, il primo valore è un intero ed i successivi numeri float). Per esempio il file potrebbe contenere i valori: 4 3.5 4.3 5.4 3.4
Scrivere un programma che legge la sequenza di float dal file serie.txt salvandoli in un array dichiarato dinamicamente V. In seguito il programma calcola il prodotto matriciale tra V (vettore colonna) e V trasposto (vettore riga) e visualizza la matrice risultante. Si utilizzino i due sottoprogrammi definiti sopra per l'allocazione e la deallocazione di matrici dinamiche di float.