Esercizi
Programmazione I e Informatica II

4 Dicembre 2015

Esercizio 1 Utilizzo della memoria dinamica

Scrivere un programma in C che esegue le seguenti istruzioni in ordine. Leggere un valore n intero da tastiera e utilizzare *calloc()* per creare un array di n variabili intere. Utilizzare un ciclo per inizializzare tutto l'array con valori presi da tastiera, utilizzando l'artimetica dei puntatori $(p+i)$. Utilizzare un ciclo per stampare l'array, questa volta utilizzando l'operatore $p[\,]$. Liberare la memoria occupata dall'array. Con *malloc* allocare un array di $2*n$ variabili intere. Stampare il contenuto di tutto l'array per vedere come é stato automaticamente inizializzato. Assegnare ad ogni elemento dell'array l'indice della sua posizione nell'array $(p[i] = i)$. Stampare nuovamente il contenuto di tutto l'array. Ridimensionare l'array da $2*n$ ad n elementi (*realloc*). Stampare il contenuto del nuovo array. Ridimensionarlo nuovamente portandolo a $2*n$ elementi. Stampare solo gli elelementi da n a $(2*n)-1$ (la seconda metà).

Esercizio 2 Artimetica dei puntatori

Dato il seguente programma, (senza implementarlo) dire se genera errori (o *warning*) al momento della compilazione.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main(void) {
6
7    int n= 0;
8
9    printf("Dammi numero elementi dell'array: ");
10   scanf("%d", &n);
11
12   int* p= calloc(n, sizeof(int)); //Quanti byte di memoria alloco qui?
13
14
15   for(int i= 0; i < n; i++) {
16     printf("Dammi elemento %d\n", i+1);
17     scanf("%d", p+i);
18   }
19
20   short* q= (short*) p;
21
22   for(int i= 0; i < 2*n; i++)
23     printf("Elemento %d: %d\n", i+1, *(q+i));
24   }
25
26   free(p);
27 }
```

Spiegare cosa come mai accade

```
Dammi numero elementi dell'array: 2
Dammi elemento 1
32768
Dammi elemento 2
4
Elemento 1: -32768
```

```
Elemento 2: 0
Elemento 3: 4
Elemento 4: 0

Dammi numero elementi dell'array: 2
Dammi elemento 1
1234567
Dammi elemento 2
3
Elemento 1: -10617
Elemento 2: 18
Elemento 3: 3
Elemento 4: 0
```

Esercizio 3 Cosa stampa?

Dato il seguente programma (senza implementarlo), scrivere cosa stampa se viene passato $n = 6$.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main(void) {
6
7  int n= 0;
8
9  printf("Dammi numero elementi dell'array: ");
10 scanf("%d", &n);
11
12
13 int* p= malloc(2*n*sizeof(int));
14
15
16 for(int i= 0; i < 2*n; i++)
17   p[i]= i;
18
19 int*  q= (int* ) p;
20
21 free(q);
22
23 int* r= calloc(2*n, sizeof(int));
24
25
26 for(int i= 0; (i < 2*n); i++)
27   if (i%2 == 0)
28     *(r+i)= i;
29
30 for(int i= 0; i < 2*n; i++)
31   if (i%2 != 0)
32     r[i]= r[i-1] + 1;
33
34 for(int i= 0; i < 2*n; i++)
35   printf("%d ", *(r+i));
36
37 puts("");
38
39 free(r);
40
41 }
42
```

Esercizio 4 Trova gli errori

Segnare quali comandi generano un errore o un *warning* a tempo di compilazione, e dire perché.

```
1 #include <stdio.h>
2
```

```c
int main(void) {

   int a= 6, b= 7;
   const int* p= &a;
   int* const q= &a;
   int* r= &b;

   *q= a+1;

   printf("a= %d\n", a);

   a= a++;

   printf("a= %d\n", a);

   *p= a+1;

   printf("a= %d\n", a);

   q= r;

   a= *r;

   printf("a= %d\n", a);

}

```

Esercizio 5 Cosa stampa?

Dato il seguente programma (senza implementarlo), scrivere cosa stampa.

```c
#include <stdio.h>

int main(void) {

   int a= 6,
   b= 7, c= 13;

   do {if (a<b) if(b%2 !=0) if(4) b++; if(c == 2*a) c++; else c--; if (a+b>=c?1:0) a--;} while(a>0);

   printf("a= %d", a);
   printf("b= %d", b);
   printf("c= %d", c);

}

```

Esercizio 6 Trova gli errori

Trova gli errori in queste dichiarazioni di variabili.

```c
int main(void) {

   float p* = null;
   int a= 3;
   double a= 3.2;
   int b= 0;
   long int B= 0;

}

```

Esercizio 7 Struttura

Creare una piccola libreria (`myLib.c`, `myLib.h`). In `myLib.h` definire la struttura

```c
struct Song { char title[64];
              char artist[32];
```

```
        char composer[32];
        short duration;
        struct Date published;
};
```

e dargli un alias *Song_t* con *typedef* (sempre in `myLib.h`). Inserire anche le dichiarazioni delle funzioni, poi definite in `myLib.h`. Queste funzioni sono

- *Song_t* allocSong(void)* crea spazio in memoria dinamica per un tipo *Song_t* e ne assegna i campi con i valori presi da tastiera (per ogni campo separatamente). Controllare se i valori in ingresso sono corretti, per esempio se *month* dentro *published* è compreso tra 1 e 12.
- *void printSong(Song_t*)*, stampa su schermo i campi delal struttura.
- *void deallocSong(Song_t* p)* libera lo spazio di memoria dinamica occupato da *Song_t*.

Ricordarsi di includere `myLib.h` in `myLib.c`. Compilare la libreria con `gcc -c mylibrary.c`. Scrivere un ulteriore file `prog.c` che contiene una funzione *main()* che utilizza in sequenza le tre funzioni. Linkare con la libreria ed eseguire prog.

Esercizio 8 Disegno figure geometriche

1. Si realizzi un programma in linguaggio C che legga un numero intero N e visualizzi un quadrato di asterischi di lato N (vedi esempio con N = 5).
2. Si realizzi una variante del programma per visualizzare solo i lati del quadrato (vedi esempio con N = 5).
3. Si realizzi una variante del programma per visualizzare un triangolo isoscele rettangolo di lato N (vedi esempio con N = 5).
4. Si realizzi una variante del programma per visualizzare un quadrato di lato N come nell'esempio del caso 4 (con N = 5).

```
Caso 1      Caso 2      Caso 3      Caso 4

*****       *****       *           *++++
*****       *   *       **          **+++
*****       *   *       ***         ***++
*****       *   *       ****        ****+
*****       *****       *****       *****
```