

**LABORATORIO DI LINGUAGGI-A**

Docente: András Horváth

Prova scritta del 02/07/2004

Serie A

COGNOME .....

NOME .....

MATRICOLA .....

Rispondere alle domande negli spazi sottostanti (o sul retro del foglio). **Non è consentito usare altri fogli.**

1. (4 punti) Cosa stampa a video il seguente programma?

Soluzione ↓

```
#include<stdio.h>

char * fun(char *x, char y, char z[])
{
    y=x[0];
    z[0]=y;
    z[1]='x';
    x[0]=z[-1];
    return x+1;
}

void main(void)
{
    char *a, b, c[]="ciao";

    b=c[2];
    a=fun(&c[0],b,&c[2]);
    printf("%c %c %s %s\n",*a,b,c,c+2);
}
```

2. (3 punti) Cosa stampa a video il seguente programma?

Soluzione ↓

```
#include<stdio.h>

void main(void)
{
    int i1=6,i2;
    do {
        i2=2;
        while(i2++<i1)
            printf("%d ",i1+i2);
        printf("\n");
    } while(i1-->=2);
}
```

3. (3 punti) Cosa stampa a video il seguente programma?

Soluzione ↓

```
#include <stdio.h>

int fun1(int x)
{
    if(x>2) return x;
    return fun1(x+3);
}

int fun2(int x)
{
    return (2*x);
}

int fun3(int x)
{
    return (x+2);
}

void main(void)
{
    printf ("%d %d %d\n", fun3(fun2(fun1(6))),fun3(fun2(fun1(1))),fun2(fun1(fun3(0))));
}
```

4. (2 punti) Scrivere delle espressioni logiche in C che corrispondono ai seguenti enunciati.

Soluzione ↓

- a) il quarto carattere della stringa puntata da `s` non è `a`
- b) `y` è maggiore di `x` ed è uguale a `z`
- c) `a` non è uguale al cubo di `c`
- d) `y` è uguale a `z` o `y` è uguale a 15

5. (4 punti) Riscrivere il seguente frammento di codice utilizzando due cicli annidati `for`.

```
int i1=4,i2;
do {
    i2=i1+1;
    while(++i2!=12)
        printf("x");
    printf("\n");
    i1+=2;
} while(i1!=10);
```

6. (4 punti) Completare il seguente programma secondo i commenti.

```
#include<stdio.h>

void main(void)
{
    int *p1,*p2,x;

    // completare la prossima riga per allocare memoria per
    // un vettore di 5 numeri di tipo int

    p1=

    // inizializzare il vettore con i valori
    // 4, 8, 12, 16 e 20 con un ciclo for

    // assegnare l'indirizzo del terzo elemento del vettore a p2

    // stampare l'indirizzo del secondo elemento del vettore

    //liberare la memoria occupata dal vettore

}
```

7. (5 punti) Scrivere, dichiarando anche le variabili necessari, un frammento di codice che

- a) apre (controllando il risultato) il file di nome "myfile" per aggiungere ulteriori righe al file
- b) aggiunge al file le seguenti righe usando cicli annidati

```
10 8 6 4 2
8 6 4 2
6 4 2
4 2
2
```

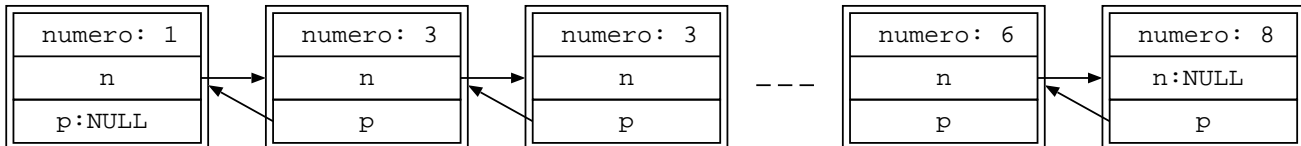
- c) chiude il file.



8. (6 punti) La seguente struttura è definita per memorizzare una serie di numeri in ordine **non crescente**:

```
struct node {
    int numero;
    struct node *n;
    struct node *p;
};
```

Un esempio dell'uso della struttura è riportato nella seguente figura:



Scrivere una funzione che ha il seguente parametro

- un puntatore di tipo `struct node` che punta al primo elemento della lista

e stampa a video tutti i numeri della catena in ordine **non crescente**. La funzione non deve restituire nessun valore.

Scrivere una funzione che ha i seguenti parametri

- un puntatore di tipo `struct node` che punta al primo elemento della lista
- un numero di tipo `int`

e cancella un nodo della lista che contiene il numero passato (se esiste un tale nodo). La funzione deve restituire un puntatore di tipo `struct node` che punta al primo elemento della lista.