

LABORATORIO DI LINGUAGGI-A

Docente: András Horváth

Prova scritta del 10/01/2005

COGNOME _____

NOME _____

MATRICOLA _____

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

1. (3 punti) Cosa calcolano le funzioni f1 e f2?

Soluzione ↓

```
char *f1(char *t)
{
    while (*(t+1) != '\0')
        t++;
    return t;
}
```

```
int f2(char *s)
{
    char *t=f1(s);
    while (s < t && *s==*t)
    {
        s++,
        t--;
    }
    if(s<t) return 0;
    return 1;
}
```

2. (4 punti) Cosa stampa a video il seguente programma?

Soluzione ↓

```
#include<stdio.h>

void main(void)
{
    int i1,i2,i3;

    for( i1=4, i2=2; i1<=10 && i2>=-8; i1++, i2-=2)
    {
        for(i3=i1+4; i1%2==0 && i3<15; i3+=2)
            printf("%d ",i3+i2);
        if(i1%2==0) printf("\n");
    }
}
```

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

Soluzione ↓

```
#include <stdio.h>

void main(void)
{
    char *citta[]={ "Roma", "Torino", "Milano", "Venezia",
                   "New-York", "Parigi", "Bologna"};
    char **p, *q, *r;

    p=citta+1;
    q=*(p+2);
    r=++q;
    printf("PRIMA STAMPA: %s\n", q+3 );
    printf("SECONDA STAMPA: %s\n", *(p+(q-r)) );
    printf("TERZA STAMPA: %s\n", ++r);
}
```

4. (3 punti) Qual è l'output del seguente frammento di codice?

Soluzione ↓

```
int V[]={1,3,5,7,9};
int *p;

printf("%d\n", V[1]);
printf("%d\n", *(V+2));
p=V+1;
printf("%d\n", *p);
printf("%d\n", *(p+2));
p=&V[1];
printf("%d\n", *p);
printf("%d\n", *p+1);
```

5. (5 punti) Implementare una procedura *non ricorsiva* void copia_senza_vocali(char *s, char *dest) che mette nella stringa dest una copia di s senza vocali (ad esempio: se s="alberto", dest="lbrt").

6. (5 punti) Scrivere una funzione `int **triangolo(int n)` che alloca dinamicamente una matrice triangolare bassa di interi con $n+1$ righe e la inizializza strutturandola come segue: la riga k contiene la sequenza dei numeri da 0 a k . Ad esempio, `triangolo(3)` deve generare la matrice

riga 0: 0

riga 1: 0 1

riga 2: 0 1 2

riga 3: 0 1 2 3

7. (5 punti) Scrivere una funzione che prende come parametri

- a) una stringa (in un vettore di caratteri),
- b) un vettore di numeri di tipo `int`,
- c) il numero degli elementi del vettore precedente (punto **b**),

e

- a) crea (controllando il risultato) un file il cui nome è passato nella stringa,
- b) scrive il numero minimo del vettore nel file,
- c) chiude il file,
- d) restituisce la media degli numeri passati nel vettore.

8. (6 punti) Sono date le seguenti definizioni, che servono a implementare alberi binari.

```
typedef struct {
    char cognome[20], nome[20];
    int qualifica;
} record;
```

```
struct nodo {
    record elemento;
    struct nodo *left, *right;
};
```

```
typedef struct nodo *tree;
```

Scrivere una funzione ricorsiva `void seleziona(tree T, tree *pT, int a, int b)` che, dato un albero `T` ordinato per qualifica, copia tutti i record di `T` per i quali $a \leq \text{qualifica} \leq b$ in `*pT`. Definire se necessario funzioni ausiliarie.