

## Fondamenti di Informatica I

PROVA SCRITTA – 8 maggio 2002

### Avvertenze:

- \* Consegnare **solo fogli formato A4**
- \* **Scrivere su un solo lato** (no fronte-retro)
- \* In ordine di preferenza usare: 1) **inchiostro nero**; 2) matita; 3) inchiostro rosso; 4) inchiostro blu
- \* **In testa a ciascun foglio** scrivere: cognome, nome, numero progressivo di pagina rispetto al totale  
esempio per il secondo foglio di 3 consegnati: Giuseppe Russo 2/3
- \* Mantenere sul banco il **libretto o altro documento di riconoscimento** fino a controllo avvenuto
- \* La **correzione** di riferimento per l'autovalutazione verrà effettuata in questa stessa aula alle ore **16:45**
- \* La consegna delle **fotocopie** dei compiti avverrà presso l'Istituto di Informatica al termine della correzione
- \* La **prova orale** si terrà **martedì 14 maggio** alle ore **9:30** presso l'Istituto di Informatica



Si ricorda che chi si presenterà all'orale **DEVE** portare l'implementazione al computer della propria soluzione, eventualmente corretta, **corredata di tutto quanto necessario alla verifica** del corretto funzionamento.

### 1. (8 punti)

Si definisca una funzione  $C$  che riceve in ingresso un vettore  $V$  di  $N$  numeri float e alloca e restituisce il vettore, di dimensioni pari al numero di elementi non negativi di  $V$ , che contiene gli indici degli elementi di  $V$  che hanno valore non negativo.

### 2. (8 punti)

Sia dato un insieme di numeri float rappresentato su una lista. L'insieme non contiene istanze multiple di uno stesso valore. Si definisca la funzione  $C$  che, ricevuto un valore, lo inserisce nell'insieme se il valore non è già presente. La funzione deve restituire FALSE se il valore da inserire era già presente nell'insieme e TRUE se invece non era presente.

### 3. (6 punti)

Si definisca la funzione  $C$  che calcola la profondità di un albero binario.

### 4. (8 punti)

Si esprimano le seguenti dichiarazioni di variabili e funzioni (prototipi) in linguaggio C:

- A** e' un array di 30 puntatori a puntatori a interi
- B** e' un array di 30 record (strutture) composti da un intero e 2 array di 25 caratteri
- C** è un puntatore a un array di 3 puntatori a interi
- D** è una funzione void con un intero e 2 puntatori a puntatori a float come parametri
- E** è una funzione che ritorna un puntatore a caratteri e ha un intero come parametro
- F** e' un array di puntatori a funzioni che ritornano un intero

```

void Set_indexes(float *V, int N, int **V_ptr) {
    int count=0, index1, index2=0;

    for(index1=0; index1<N; index1++)
        if(V[index1]>=0) count++;
    *V_ptr=(int*)malloc(count*sizeof(int));

    for(index1=0; index1<N; index1++)
        if(V[index1]>=0) {
            *V_ptr[index2]=index1;
            index2++;
        }
}

```

```

.....

struct list { float value; struct list *next_ptr; };

Boolean insert(struct list **list_ptrptr, float value) {
    Boolean not_found=TRUE;

    while(*list_ptrptr!=NULL && not_found==TRUE)
        list_ptrptr=&(*list_ptrptr)->next_ptr;

    if(not_found==TRUE) {
        *list_ptrptr=(struct list *)malloc(sizeof(struct list));
        (*list_ptrptr)->value=value;
        (*list_ptrptr)->next_ptr=NULL;
    }
    return not_found;
}

```

```

.....

struct btree {int value; struct btree *lx; struct btree *rx; };

int depth(struct btree *root) {
    if(root==NULL) return 0;
    else return 1 + max( depth(root->lx), depth(root->rx) );
}

```

```

int ** A [30];
struct { int a; char b[25]; char c[25]; } B [30];
int (* C) [3];

```

```

void D (int, float **, float **);
char * E ( int );
int (* F [ ] ) ();

```