

## Fondamenti di Informatica A-L – (Prof. P. Zingaretti)

PROVA SCRITTA – 23 giugno 2003

### 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
- \* Nient'altro deve trovarsi sul banco: **non è consentito consultare libri, dispense, appunti, ecc.**
- \* La **correzione** di riferimento per l'autovalutazione verrà effettuata in questa stessa aula alle ore **15:45**
- \* La consegna delle **fotocopie** dei compiti avverrà presso l'Istituto di Informatica al termine della correzione
- \* La **prova orale** si terrà **giovedì 3 luglio** alle ore **9:00** 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. (10 punti)

Si definisca una funzione C (Pascal) che riceve in input una matrice di interi con N righe e M colonne (N e M qualsiasi) e restituisce un vettore di M elementi in cui il valore di ogni elemento è dato dalla somma degli elementi sulla colonna M della matrice.

### 2. (10 punti)

Si definisca una funzione C (Pascal) che restituisce la rappresentazione esadecimale di un numero ricevuto in input.

### 3. (10 punti)

Si scriva una funzione C (Pascal) che inserisce un intero su una lista dinamica (rappresentazione collegata mediante puntatore) in modo che, finché possibile, alterni numeri pari a numeri dispari.

### 1. (10 punti)

Si definisca una funzione C (Pascal) che riceve in input una matrice di interi con N righe e M colonne (N e M qualsiasi) e restituisce un vettore di M elementi in cui il valore di ogni elemento è dato dalla somma degli elementi sulla colonna M della matrice.

```
int *less_ave(int *MAT, int N, int M) {
    int i, j, *V=(int *)malloc(M*sizeof(int));

    for (i=0; i<M; i++) {
        V[i]=0;
        for (j=0; j<N; j++) V[i]+=(MAT+j*M+i);
    }
    return V;
};
```

### 2. (10 punti)

Si definisca una funzione C (Pascal) che restituisce la rappresentazione esadecimale di un numero ricevuto in input.

```
char *esa(int N) {
    int i=0, j, R, S[8];
    char *s;
    do {
        R=N%16;
        S[i++]=R;
        N/=16;
    } while (N);
    s=(char *)malloc((i+1)*sizeof(char));
    for (j=0; j<i; j++)
        s[j]=S[i-1-j]+(S[i-1-j] > 9 ? 'A'-10:'0');
    s[j]='\0';
    return s;
}
```

### 3. (10 punti)

Si scriva una funzione C (Pascal) che inserisce un intero su una lista dinamica (rappresentazione collegata mediante puntatore) in modo che, finché possibile, alterni numeri pari a numeri dispari.

```
typedef struct list_tag { int value; struct list_tag *next; } list;
void ins_pari_dispari(list **lpp, int value) {
    list *tmp_lp;
    int pari=0, dispari=0, found=0;

    while ((*lpp!=NULL) && !found) {
        if ((*lpp)->value%2) {pari=0; dispari++;}
        else {pari++, dispari=0; }
        if (pari<2 && dispari<2) lpp=&((*lpp)->next); //posizionamento
        else found=1;
    }
    tmp_lp=*lpp; //salvo resto della lista
    *lpp=(list *)malloc(sizeof(list));
    (*lpp)->value=value; //inserimento
    (*lpp)->next=tmp_lp; //ricollego
}
```