

```

/* 22 gennaio 01    Esame scritto, Fondamenti Informatica I
   cdl Ingegneria Elettronica, Universita' di Ancona
*/

/*
(8 punti)
Si scriva la funzione c (Pascal) che calcola la media dei valori memorizzati
su un albero binario di numeri float (real).
*/

#include<stdio.h>
struct btree{
    float value;
    struct btree * lx_ptr;
    struct btree * rx_ptr;
};

void get_sum_and_size(struct btree * root_ptr, float * sum_ptr, int * size_ptr);

float average(struct btree * root_ptr)
{
    float sum;
    int size;

    size=0;
    sum=0.0;
    get_sum_and_size(root_ptr, &sum, &size);

    return sum/size;
}

void get_sum_and_size(struct btree * root_ptr, float * sum_ptr, int * size_ptr)
{
    if(root_ptr!=NULL)
    {
        (*sum_ptr)+=root_ptr->value;
        (*size_ptr)++;
        get_sum_and_size(root_ptr->lx_ptr, sum_ptr, size_ptr);
        get_sum_and_size(root_ptr->rx_ptr, sum_ptr, size_ptr);
    }
}

```

```

/* (10 punti)
E' assegnata una sequenza A di valori float (real) rappresentati su una lista. Si
assuma che la sequenza sia ordinata in ordine decrescente (il primo elemento e' il
massimo). Si vogliono costruire due sequenze A1 e A2 che partizionano A in maniera
da minimizzare la differenza sulla somma dei valori contenuti in A1 e A2.
(Per costituire una partizione, qualsiasi valore di A deve essere rappresentato su
una e una sola delle due sottosequenze, e viceversa qualsiasi valore contenuto in
alcuna delle due sottosequenze deve essere contenuto A).
Si consideri il seguente algoritmo approssimato e se ne dia una rappresentazione
come funzione del c (Pascal): per ogni elemento x della sequenza A, calcola la
somma dei valori nelle sottosequenze A1 e A2, e inserisci il valore x nella
sottosequenza con somma minore; nel caso che A1 e A2 abbiano la stessa somma,
inserisci il valore in una qualsiasi delle due.
(Suggerimento: si scriva la soluzione in forma iterativa.)
*/

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

void insert(struct list ** list_ptrptr, float value);

void partition(struct list *A_ptr, struct list **A1_ptrptr, struct list **A2_ptrptr)
{
    float sum_A1;
    float sum_A2;

    sum_A1=0;
    sum_A2=0;
    while(A_ptr!=NULL)
    {
        if(sum_A1<=sum_A2)
        {   insert(A1_ptrptr,A_ptr->value);
            sum_A1+=A_ptr->value;
        }else
        {   insert(A2_ptrptr,A_ptr->value);
            sum_A2+=A_ptr->value;
        }
        A_ptr=A_ptr->next;
    }
}

void insert(struct list ** list_ptrptr, float value) /* solito inserimento */
{
    while(*list_ptrptr!=NULL)
        list_ptrptr=&((*list_ptrptr)->next_ptr);

    (*list_ptrptr)=(struct list *)malloc(sizeof(struct list));
    (*list_ptrptr)->value=value;
    (*list_ptrptr)->next_ptr=NULL;
}

```

```
/* (6 punti)
Si scriva la funzione c (Pascal) che riceve in ingresso un vettore di float (real) V
e un vettore di indici Index, e che alloca e costruisce un vettore SubV il quale
contiene solo gli elementi di V nelle posizioni corrispondenti agli indici di Index.
Si assuma che Index sia una sottosuccessione degli indici degli elementi di V, ovvero
che esso abbia dimensione minore o uguale a quella di V, e che sia un vettore
ordinato e senza ripetizioni di valori naturali minori della dimensione di V.
(Ad esempio, se V={17.3, 15.1, 18, 2.4, 7.9} e Index={0,2,3}, allora
SubV={17.3, 18, 2.4}.) */
```

```
void sample(float * V, int * Index, int Index_size, float ** subV_ptr)
{
    int count;

    *subV_ptr=(float *)malloc(Index_size*sizeof(float));
    for(count=0;count<Index_size;count++)
    {
        *subV_ptr[count]=V[Index[count]];
    }
}
```

```
/* (6 punti)
Si scriva il frammento della sintassi delle espressioni del c
che permette di interpretare l'espressione x=2*(y++)+(z<10).
Usando tale sintassi si costruisca l'albero sintattico, e si usi tale
albero per valutare il valore restituito dall'espressione nel caso in
cui inizialmente y vale 10, e z vale 12*/
```

```
/*
    expr ::= const | var | expr op2 expr | var++ | (expr)
    op2 ::= * | + | <
*/
```