

In C, all variables must be declared before they are used, usually at the beginning of the function.

(ב) תמ"ז' נבו. ~~החל~~ ~~פאס~~ עכ"ל EOF-80 even
עצמי סוף קרובים/קטן ואכן כ"ח שנה מאז המיון
~~הוא~~ אין הנהגה לערכו. (זה 152 בספר, חומרת הנאשם)

(ח) האלה לא ~~היה~~ נבונה. הידוע שערכות הנהגות
 יש המבטות על עבודה עם קבוצים אם מסדר תוכניות
 במתקנים, למשל ב"כ לא ניתן לשתות את אותו קופסא שכתבה
 מ-2 תוכניות. אם התוכנית פתחה קופסא ועושה בו
 שימוש מיוחד, הקופסא ~~היא~~ יחד נעלם צד שהתוכנית חזק
~~היא~~ כמסדר, הקופסא נאמר נעלם גם בזמן שהתוכנית הנהגת
 לא משתמש בו, ומונעת שימוש בו ע"י תוכניות אחרות.
 • קיימות המבטות נוספות א' שערכות הנהגות ושכן כנ"ל
 מסביר את הקופסא כמא המשתמש בו.

~~you are not allowed to have any more of~~

סלסה 2

(6)

```
#define count(x, c) \
{\
    int i;\
    x >>= 1;\
    for(i=0; i < (sizeof(x)/2); i++) \
    {\
        if(x & 1) ++c;\
        x >>= 2;\
    }\
}
```

~~התשובה~~

המאקרו מקבל 2 פרמטרים. אריסתה מאותו ערך, ערכו, אריסתה
 אריסתה שיהיה את התשובה.
 התחנה מקבלת x ו shift ימין כדי שיהיה
 שני המאקרו יבנה הביטוי הכי ימני.
 רש המאקרו for ובזמן בעצמו מסלק AND הא
 הביטוי הכי ימני נוסף. אחרי הבזקה מקבלת shift ימין.


```
#include <stdio.h>
```

2 ндк

(

```
#define count(x,c) \
```

31/

```
int i; \
```

$$x \gg 1; 1$$

```
for (i=0, c=0; i < (sizeof(x)/2)*8; ++i) \
```

31

$$if(x \neq 1) ++c; \setminus$$
 $x > 2; \quad \backslash$

21

31

```
void main()
```

3

```
int res[10], arr[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

```
int i; temp;
```

```
for (i=0; i<10; ++i)
```

5

```
temp = arr[i];
```

count(tmp, res[i])

22

3

התוכנית נועדה על העדף ~~ל~~ זר, עבור כל הא
עתיקה אותו למסורה צמי (כי המאקרו ~~לא~~
אם הביטוי שנת מקבל) ומפיעה את המאקרו על המסורה
הציוני והמא המאזי ביעוד התשובות res.

4ce

(1)

(2)

typedef struct tnode

{

int node_num;

char * node_name;

struct tnode * left;

struct tnode * right;

} tree_node;

4ce

(1)

print-tree

(2) ממש הבדק

(3) ע"י איוש הפונקציה print_tree והפונקציה
השמה הפנימית:

num=4 name=name4

num=2 name=name2

num=4 name=name4

num=5 name=name5

num=8 name=name8

num=9 name=name9

num=6 name=name6

num=3 name=name3

num=1 name=name1

16
הפונקציה

12 4 35 7 6 8 9


```
#include <ctype.h>
```

4.2.8ce

```
#include <stdio.h>
```

10 10

```
#include <string.h>
```

2

```
typedef struct tnode {
```

```
    int num;
```

```
    char * name;
```

```
    struct tnode * left;
```

```
    struct tnode * right;
```

```
} tree_node;
```

```
tree_node *new_node(int num, char * name)
```

```
{
```

```
    tree_node *p;
```

```
    p = (tree_node *) malloc (sizeof (tree_node));
```

```
    p->name = (char *) malloc (sizeof (char) * 10);
```

```
    p->left = NULL;
```

```
    p->right = NULL;
```

```
    p->num = num;
```

```
    return p;
```

```
}
```

```
void print_tree (tree_node * p)
```

```
{
```

```
    if (p->left != NULL) print_tree (p->left);
```

```
    if (p->right != NULL) print_tree (p->right);
```

```
    printf ("num=%d, name=%s\n", p->num, p->name);
```

```
}
```

~~void add_tree (tree_node * n)~~

~~{~~

~~n->left = NULL;~~

~~n->right = NULL;~~

~~if (n->left == NULL)~~

~~add_tree (n->left);~~

~~if (n->right != NULL)~~

~~add_tree (n->right);~~

~~}~~

~~void add_tree (tree_node * n)~~

~~{~~

~~if (n->left == NULL) n->left = n;~~

~~if (n->right == NULL) n->right = n;~~

void add_tree (tree_node * n)

{

if (n->left == NULL) {

n->left = p;

return; }

else if (n->right == NULL) {

n->right = p;

return; }

~~add_tree~~

else {

add_tree (n->right);

}


```

void main()
{
    char *names[10] = {"name1", "name2", "name3", "name4", "name5",
                       "name6", "name7", "name8", "name9"};
    tree_node *root, *new;
    int i

    for (i = 1; i <= 10; ++i)
    {
        new = new_node(i, names[i-1]);
        add_node(new);
    }
    print_tree(root);
}

```

על המסמך להכין את חקיות בניית העץ
על המסמך על אזור