

עמוד 1

(א) כפי שכתבנו בעבר בדף 41:

"The % operator cannot be applied to float or double"

כלומר, לא ניתן להשתמש באופרטור % עם מסוגי הנתונים float ו-double. הסיבה לכך היא שהאופרטור % מיועד למסוגי הנתונים int, long וכו'.

(ב) להלן דוגמה קטנה:

הבעיה היא ש-C const אינו יכול להכיל משתנים. לכן, כדי להשתמש ב-const, עלינו להשתמש ב-#define. לדוגמה, במקום לכתוב:

```
const int MAX = 100;
```

עלינו לכתוב:

```
#define MAX 100
```

הסיבה לכך היא ש-C לא תומך במשתנים const. במקום זאת, הוא משתמש ב-#define כדי להגדיר קבועים. זהו דבר חשוב לזכור כשאתם עובדים עם C.

ב-C, ישנן שתי דרכים להגדיר קבועים: באמצעות const או באמצעות #define. הדבר הראשון שיש לדעת הוא ש-const אינו תומך במשתנים, בעוד ש-#define תומך בהם. לכן, אם אתם רוצים להשתמש במשתנים, עליכם להשתמש ב-#define.

~~הדוגמה הבאה מראה כיצד להשתמש ב-#define כדי להגדיר קבועים. בדף 41, ראינו כיצד להשתמש ב-const. הדבר הראשון שיש לדעת הוא ש-const אינו תומך במשתנים, בעוד ש-#define תומך בהם. לכן, אם אתם רוצים להשתמש במשתנים, עליכם להשתמש ב-#define.~~

הדוגמה הבאה מראה כיצד להשתמש ב-#define כדי להגדיר קבועים.

(ג) להלן דוגמה קטנה:

```
int main() {
    int x = 10;
    printf("x = %d", x);
    return 0;
}
```

הדוגמה הזו מראה כיצד להשתמש ב-printf כדי להדפיס את ערך המשתנה x. הדבר הראשון שיש לדעת הוא ש-printf הוא פונקציה שמוצאת מקורה בספריית stdio.h. לכן, עלינו לכלול את הספרייה הזו בתוכנית.

(5)

הדוגמה הזו מראה כיצד להשתמש ב-printf כדי להדפיס את ערך המשתנה x. הדבר הראשון שיש לדעת הוא ש-printf הוא פונקציה שמוצאת מקורה בספריית stdio.h. לכן, עלינו לכלול את הספרייה הזו בתוכנית.

2 side



4

1



1

[illegible]

1

•

1

I

[illegible]

אם נניח כי α הוא מספר ממשי, אז α הוא מספר רציונלי.

10. 11. 2011 12. 11. 2011 13. 11. 2011 14. 11. 2011 15. 11. 2011 16. 11. 2011 17. 11. 2011

2011. 6. 28. 22:00. 6. 28. 22:00 - 22:00. 6. 28. 22:00

[illegible]

ה'תשנ"ח אדר א' י"ב יום ישיבה נמרצת

הנהגה נכונה של הרכיבים הנ"ל תורמת להגדרת המערכת.

Let the price of a unit of good 1 be 1 and the price of a unit of good 2 be p .

ענין זה אינו נכון

```
#include <stdio.h>
#define SIZE 8
#define ZEROS(A,B,ANS,ANSCOUNT) {
    int mask = 1;
    int counta = 0;
    int countb = 0;
    while (mask != 0) {
        if (!(A & mask))
            counta += 1;
        if (!(B & mask))
            countb += 1;
        mask <<= 1;
    }
    if (counta > countb) {
        ANS = A;
        ANSCOUNT = counta;
    }
    else {
        ANS = B;
        ANSCOUNT = countb;
    }
}

int main() {
    int A, B, ANS, ANSCOUNT;
    printf("Enter two integers: ");
    scanf("%d %d", &A, &B);
    ZEROS(A, B, ANS, ANSCOUNT);
    printf("The result is: %d\n", ANS);
    return 0;
}
```

main
ZEROS


```

void main (void) {
    int arr[SIZE];
    scanf ("%d %d %d %d %d %d %d %d",
            ,arr[0],arr[1],arr[2],arr[3],arr[4],
            ,arr[5],arr[6],arr[7]);
    int i, max=0, maxcount=0;
    for (i=0; i<SIZE-1; i++)
        ZEROS (arr[i],arr[i+1],max,maxcount);
    printf ("The number: %d Its count: %d",
            max,maxcount);
}

```

הפונקציה ZEROS מקבלת שני מספרים שלמים ומוציאה את המספר הגדול יותר ואת מספר הופעותיו. הפונקציה מקבלת שני מספרים שלמים ומוציאה את המספר הגדול יותר ואת מספר הופעותיו. הפונקציה מקבלת שני מספרים שלמים ומוציאה את המספר הגדול יותר ואת מספר הופעותיו.

שאלה 3

(א) הקטע שזו

ישנה שגיאה בשורה 4 מכיוון שבשורות 1-3
 realloc לא מוגדרת לפני כי לא ידוע מהספיק זיכרון
 לקבל לפני היא חלשה להסיר NULL דבר יוצא
 ויש להו בודקים במקרה שזו לא נמצא להסיר לכתוב
 להקטע.

הז'ר נוספת:

הסר free אלא טורה 13

(-4)

(ב) קטע הקוד שזו, בהדפסה במקרה הנ"ל נכתב:

$d = 0$ במקום $d = 1$. כפי לקחו שגור

ישו יבואים להגידה שר האשגיה מקובלים:

לפיה: $d = 1$; $a = b = c = 1$;

אז אחריון אחריון שר בהדפסה אלה הסוד

ברוך אהמלה אלא (שורה הקוד) ה'רן קרן

(ג) קטע הקוד שזו במקרה יוצא ישו הזכרנו

(MYEXIT) חסרים סוגריים אשר יקבלו מר

אשר בהדפסה וה'רן. מתקנת, אלא סוגריים אלא

ישו נצא מתקנת. אלא קולט האמה אלא קר

ישו חסר " שליו

כדי לקחו שר " להסיר סוגריים להסירה

אשר יקבלו שר שר הקוד

(-4)

הז'ר נוספת: ס'לם גמ'לם א' נפון

4 ake



```

type def supplier *sapliers
typedef struct suppliers {
    int id ;
    int duration ;
    float debt ;
    struct supplier * next
} supplier

```

④ חלון מדג'רים בחן

[illegible]

קריאה: (עמוד 4 דף 2) יו/מזל זה באופן קבוע

הקוד הוא המנהל מזהה בעמוד 1.

add Supplier - פונקציה אשר נמצאת בעמוד 105. פונקציה זו
 עושה קריאה אל וזיכרון רשימת מזהים בלתי אחרונה ריקה.
 הכנסה של המספר הרשימה המבוצעת תוך כדי עריכה של המספר
 המספר של הרשימה. אלן יוצרים מנהל מספר מזהים במידה
 ובמשימה ריקה הוא נחשב הרשימה. אם לא ריקה אלן המשימה
 תיבן להוסיף מזהים ומחלקים אלן עם המספר המספר יי
 מזהים המספרים.

compare - פונקציה אשר נמצאת בעמוד המבוצעת אל המספרים
 המספר המספר של רשימה ומספרים 2 אל המספרים
 יש מזהים מזהים ! 0 אל המספרים
 קודם כל אלן המספרים המספר המספרים ריקה אלן
 עניין ריקה במספרים אלן המספרים אלן המספרים
 המספרים אלן המספרים אלן המספרים אלן המספרים
 המספרים אלן המספרים אלן המספרים אלן המספרים
 המספרים אלן המספרים אלן המספרים אלן המספרים
 המספרים אלן המספרים אלן המספרים אלן המספרים

main - המספר המספר של המספרים והמספרים
 המספרים אלן המספרים אלן המספרים אלן המספרים
 המספרים אלן המספרים אלן המספרים אלן המספרים

(after 2111 →) 4 die

#define COUNT 10

(2)

typedef struct supplier *

supplier;

int id;

int duration;

float debt;

struct supplier *next;

} supplier;

~~supplier *~~ supplier * addSupplier(supplier *sup, int id, int dur, float debt)

{
supplier new, cur, prev;

new = (supplier *) malloc(sizeof(supplier));

if(!new) return NULL;

new->id = id; new->duration = dur; new->debt = debt;

new->next = NULL;

if(sup == NULL) return new;

prev = sup; cur = sup->next;

while((cur != NULL) && (cur->id < new->id)) {

prev = prev->next; cur = cur->next; }

if(cur == NULL)

prev->next = new;

else {

prev->next = new;

new->next = cur; }

return sup;

}

לשימוש הבודק

כיון מוגדר

כיון

```
int compare (supplier * sup1, supplier * sup2) {
```

```
    int isIdentical=1, isCorolating=0;
```

```
    if (sup1 == NULL && sup2 == NULL) return 2;
```

```
    if (sup1 == NULL || sup2 == NULL) return 0;
```

```
    supplier cur1, cur2;
```

```
    while (cur1 != NULL && cur2 != NULL) {
```

```
        if (cur1 == cur2) {  

            if (isCorolating == 0) {  

                isCorolating = 1;  

        if ((find (cur1, cur2->id) || find (cur2, cur1->id)) && isCorolating == 0)  

            isCorolating = 1;
```

```
        if (cur->id != cur2->id && isIdentical == 1)
```

```
            isIdentical = 0;
```

```
        cur1 = cur1->next; cur2 = cur2->next; }
```

```
    if ((cur1 != NULL || cur2 != NULL) && (isIdentical))
```

```
        isIdentical = 0;
```

```
    if (isIdentical) return 2;
```

```
    else if (isCorolating) return 1;
```

```
    else return 0; }
```

```
void main (void) {
```

```
    suppliers list1 = NULL, list2 = NULL; int i;
```

```
    for (i=0; i<COUNT; i++) {
```

```
        list1 = addSupplier (list1, i, i*30, i*50.5);
```

```
        list2 = addSupplier (list2, i, i*30, i*50.5); }
```

```
    int result = compare (list1, list2);
```

```
    if (result == 2)
```

```
        printf ("Lists are equal");
```

```
    if (result == 1)
```

```
        printf ("Lists not equal but corolating");
```

```
    else
```

```
        printf ("different lists !!");
```

```
}
```

1.3A) כל האנשים
הם מוגדרים

הם מוגדרים
הם מוגדרים


```

int find(supplier *sup1, int id) {
    if (sup1 == NULL) return NULL;
    while (sup1 != NULL)
    {
        if (sup1->id == id)
            return 1;
        sup1 = sup1->next;
    }
    return 0;
}

```

@ הגיון
 מוכיח
 אם קיים
 אם לא קיים
 1
 0
 1-8