מלא את הפרטים בכל המקומות הדרושים

מספר תעודת זהות (9 ספרות) | מספר הקורס: 20465

הדבק כאן את מדבקת הנבחן

לשימוש הבודק

שאלה 1:

א. נכון, המחרוזת שנוצרת היא כמס' ה-char-ים +1 בלבד סוף מחרוזת "\0". לפי עמ' 38 בספר: "a string constant is an array of characters. The internal representations of the string has a NULL character '\0' at the end".

ב. נכון, כאשר מגדירים את המערך a[i] נוצר הצב' של i פרמטרים בקבר, כאשר נקרא a האיבר הראשון בתוכתו היא הראשונה בהצב' האיברים לפי שרתים בספר בעמ' 28: "when the name of the array is used as an argument the value passed to the function is the location or address of the beggining of the array". כך שכשמעבירים את שם המערך a לפונקציה בפרמטר הלהבינו את כתובת של תחילת המערך.

ג. נכון, קריאה מערכת (Systemcall) read מאפשר קריאה של מספר כמו שרוצה בספר בעמ' 70 המלוווה: "Any number of bytes can be read or written in one call... Larger sizes will be more efficient because fewer systemcalls will be made".

ואכן, מומלץ לקרוא כמה גדולה יותר של מספר בקריאה אחת מאשר הרבה קריאות של מעט בתים כי אז נחסך זמן יותר של קריאות עשיה המערכת (עם אל שנפול סיב הוא הקריאה לא אחת בכל קריאה)

שאלה 2:

לשימוש הבודק

```
#include <stdio.h>
#include <stdlib.h>
#define find_bits(X, Y)\
{\
int i;\
double zero=0, one =0 ;\
for(i=0; i<((sizeof(x)*8); i++){\
    if((x&y >>(i))<<(i) == 1){\
        one++}\
    else{\
        zero++}\
if( one /(zero+one)>=0.6)\
    Printf("the 2 variables has at least 60%- match bits")\
}   Printf( %d, %d, x, y);\
int main()
{
int array[5]={11,32,43,74,15};i;
int iparam=93;
for(i=0; i<(sizeof(array[]/sizeof(array[0])); i++)
{
  find_bits(iparam, array[i]);
}
return (0);
}
```

יש להגדיר את הקוד מה main לפונקציה.
בסעיף ב' נדרשה פונקציה, שה-main
מפיה וקורה ולא מפיק מערך ולהדר.

שאלה 3:

ג. בתכנית הבאה קיימת שגיאה, תנאי ה- if (בשורה 7 ✓) הזה לעולם לא יתקיים ולכן לעולם לא יודפס good night. ניתן לשנות את האופרטור || (or) לאופרטור && (and) כך שבעבור הקריאה f(MESSAGE3) יודפס פעם אחת בלבד good night.

ב. בפונקציה קיימת שגיאה, בלולאת ה- for בשורה 5 צריך לשנות את תנאי העצירה ל- i<9 (קטן ממש מ-9) ובשורה 6 לשנות את תנאי העצירה ל- j<11 (קטן ממש מ-11) מכיוון שאם גודל המטריצה מוגדר ל- 9x11 אז האינדקס הגבוה של i יהיה 8 (0 עד 9) והאינדקס הגבוה של j יהיה 10 (0 עד 10).
הערה נוספת היא שהפונקציה בשום שלב לא מדפיסה את האיברים (כנתון בשאלה "אמורה להדפיס דוגמת דגל מסוים...") ניתן לפתור זאת ע"י הוספה של שורה נוספת אחרי שורה 7 בתוך הלולאות של המטריצה.

```
printf("[%d][%d] = %c", i, j, stars[i][j]);
```

א. ההדפסה בשורה האחרונה תמיד תהיה 0, מכיוון שדבר ראשון ה count הוא מאפס את total ורק לאחר מכן בודק האם buf הוא NULL. ניתן לשנות בשורה 7 כך שרק נגדיר את total ולא נאפס אותו (static int total;), בכל מקרה המגיעה ו buf איננו NULL ה- if בשורה 10 יתקיים ושם total מתאפס מחדש ו- ח שומר אין ערכו הקודם.
בנוסף, לולאת ה- for בשורה 20 שרצה כל עוד קיימים איברים במערך מבצעת קריאה כל פעם ל count אך לא מקבלת מ count שום ערך וגם לא מדפיסה אותו, צריך להוסיף בשורה 23 Printf(the count is %d\n", count(s,'a'));
אם לא נוסיף את השורה זו כל קריאה ל- count נאפס את התוצאה הקודמת ולעולם לא תודפס תוצאה שונה מ-0.

שאלה 4:

לשימוש הבודק

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

typedef enum target{ shivuk, news, forum, other};
typedef struct webs* ptr;
typedef struct webs{
        int num;
        int enters;
        target tar;
        char* webname;
        char* url;
        ptr next;
} website;
ptr head1=NULL;
Ptr head2=NULL;

Void readFILE(FILE *file, Ptr *head)
{
char buf[200];
ptr temp;
while(!eof(file)){
if(!fgets(buf,200,file)){
    continue;
}
temp=(ptr)(malloc(sizeof(web)));
if (!temp) exit(1);
char *p;
p=strtok(buf," ");
temp->num=atoi(p);
p=strtok(NULL," ");
temp->enters=atoi(p);
```

330479

לשימוש הבודק

```
P=strtok(NULL, " ");
temp->tar =atoi(p);
P=strtok(NULL, " ");
strcpy(temp->webname,P);
P=strtok(NULL, " ");
strcpy(temp->URL,P);
add(temp,head)
}
Void add(ptr web, ptr head){
if (head!= NULL)
{
  web->next =head;
  head=web;
}
else{
    head=web;
    web->next=NULL;
}
Ptr head3=NULL
Ptr * Combine(ptr *head1, ptr *head2){
 While((head1)&&(head2)){
   if (head1->num >head2->num){
     Ptr P1, P2;
     P1=head1->next;
     P2=head2->next;
     head1->NEXT=head3;
     head2->NEXT=NULL;
     head3=head1
     P1=P1->NEXT;
     P2=P2->NEXT;
}
```

לשימוש הבודק

```
if (head1 -> num < head2 -> num){
    P1 = head1 -> NEXT;
    P2 = head2 -> NEXT;
    head2 -> next = head3;
    head1 -> NEXT = NULL;
    head3 = head2;
    P1 = P1 -> NEXT;
    P2 = P2 -> NEXT;
}
if (!head1){
    head3 -> NEXT = head2;
}
if(!head2){
    head3 -> NEXT = head1;
}

Void freelist(ptr *head)
{
  Ptr p;
  While(*head != NULL)
  {
    P = *head;
    *head = (*head) -> next;
    free(P)
  }
}

int main()
{
ptr head1, head2;
readfile(filelist1, &head1)
readfile(filelist2, &head2)
Combine(&head1, &head2);
return(0);
}
```