שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>
void main()
{
     printf("%d",printf("%d",printf("%d",1)));
}
```

יש לנמק את התשובה.

111

פונקציה printf פנימית מדפיסה 1 ומחזירה מספר בתים כתובים שווה ל-1. לכן printf שנייה מדפיסה 1 וגם מחזירה 1 המודפס ע"י printf חיצונית.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>
#define SIZEOF -

void main()
{
     printf("%d",EOF SIZEOF (int)sizeof EOF);
}
```

יש לנמק את התשובה.

-5

ערך מספרי של EOF שווה 1-. זה מספר שלם וגודלו הוא 4 בתים. מילה SIZEOF מוחלפת בסימן – (מינוס). לכן מקבלים 5-=4-1-

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>
void main()
{
     int i;
     for (i=0; i<sizeof"sizeof"[i]; i++)
          printf("%d",sizeof"sizeof"[i]);
}
```

יש לנמק את התשובה.

1

"sizeof"sizeof זה גודל של תו מס' i במחרוזת "sizeof". גודל של כל תו הוא 1 (בית אחד). לכן באיטציה הראשונה i=0 והתנאי 0<1 מתקיים. נכנסים לגוף הלולאה ומדפיסים 1. באיטרציה השנייה i=1 והתנאי 1<1 לא מתקיים. יוצאים מהלולאה.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>
void main()
{
     char *s="ab";
     printf("%d\n",sizeof s[0]+sizeof s[1]-sizeof'a'-sizeof'b');
}
```

יש לנמק את התשובה.

-6

גודל של משתנה מסוג char הוא 1. תו קבוע הוא פשוט ייצוג של מספר שלם בקוד ASCII שלו וגודל של קבוע שלם הוא 4. לכן 1+1-4-4=-6 .

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
  char *s="123";
  puts( (char *)( ((long long)s*(long long)s+2*(long long)s+1)/(long long)(s+1) ) );
}
```

יש לנמק את התשובה.

23

<u>הסבר:</u>
Casting ל- long long ממיר כתובת (ערך s) למספר שלם.
מקבלים

$$(s^2+2s+1)/(s+1)=(s+1)^2/(s+1)=s+1$$

Casting ל- * char קובע ערך s+1 ככתובת של תו מס' 1 במחרוזת.
פלט מתו מס' 1 עד סוף המחרוזת:

23

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
  int a[100];
  printf("%d", sizeof a + sizeof sizeof a + sizeof sizeof sizeof a);
}
```

יש לנמק את התשובה.

408

sizeof a זה גודל של מערך a והוא 400 בתים (100 איברים מסוג int, גודל של int הוא 4 בתים).
sizeof sizeof a זה גודל של 400 אשר טיפוסו int ולכן הוא 4 בתים.
sizeof sizeof sizeof a זה גודל של 4 אשר טיפוסו int ולכן הוא 4 בתים.
400+4+4=408

חלק 3.

שאלת בונוס (10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
  char C[]="C++", c;
  for (c=0; c<3; c++)
       C[c]+=!c?'+'-'+':'-'-'+';
  puts(C);
}
```

חובה לנמק את התשובה! התשובה ללא נימוק או עם נימוק לא נכון לא תזכה בנקודות.

C--

ל- c==0 מתקיים c! ולכן מבצעים פעולות לפני ":" ל- C[0] מחברים קוד ASCII של '+' ומחסירים קוד ASCII של '+'. לכן C[0] לא משתנה וערך שלו נישאר 'C'. ל- c==1 ו- c==2 לא מתקיים c! ולכן מבצעים פעולות אחרי ":" ל- C[1] ו- C[2] מחברים קוד ASCII של '-' ומחסירים קוד ASCII של '+'. ערך של התווים הוא '+' ולכן אחרי חיסור '+' וחיבור '-' ערך שלהם משתנה ל- '-'.

שאלת בונוס (10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
      int n = strlen("\\\\"), m = strlen("\n\n\0\n"), k = strlen("////");
      char *s = "Ah!", *x = s + m;
      puts( (char *)( (long long)(pow((int)x, n) - k) / (int)(x + m) ) );
}
```

חובה לנמק את התשובה! התשובה ללא נימוק או עם נימוק לא נכון לא תזכה בנקודות.

Ah!

סדרת escape שהתו הראשון שלה הוא '\' מציגה לא את עצמה אלא איזה תו אחר. במיוחד '\n' זה תו של סוף השורה, '\0' זה תו של סוף המחרוזת, '\\' זה תו '\' עצמו. לכן מחרוזת "\\\\" מורכבת מ-2 תווים וארכה הוא 2. ב- "\n\n\0\n" יש שני תווים עד לסוף המחרוזת ואורכה 2. מחרוזת "////" מורכבת מתווים רגילים ואורכה 4. אזי n=2, m=2, k=4 ו- x=s+2.
Casting ל- long long ול- int ממיר כתובת (ערך x) למספר שלם.
$(x^2-4)/(x+2)=(x-2)(x+2)/(x+2)=x-2=s+2-2=s$
Casting ל- * char קובע ערך s ככתובת של התו הראשון במחרוזת.
לכן puts מקבלת מחרוזת s ומדפיסה את התוכן שלה "Ah!".

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
     printf("%d\n", sizeof"|0\0/00\0/0|");
}
```

יש לנמק את התשובה.

11

בטקסט שבתוך המירכאות יש 12 סימנים. ביניהם פעמיים מופיעה סדרת '\0' אשר מהווה תו אחד. לכן מופיעים פה 10 תווים. בסוף כל מחרוזת קבועה יש עוד תו אפסי ('\0'). לכן סה"כ המערך מורכב מ-11 תווים. אופרטור sizeof מחזיר את גודל האופרנד בבתים. כל תו תופס בית אחד. לכן הפלט יהיה 11.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>
void main()
{
     printf("%d",printf("%d%d",printf("%d",11),printf("%d",11)));
}
```

יש לנמק את התשובה.

1111222

כל אחת מהפונקציות printf("%d",11) מדפיסה 11, סה"כ 1111. פונקציית printf מחזירה מספר בתים כתובים. לפן כל אחת מהפונקציות הנ"ל מחזירה 2. אז printf שלישית (מימין) מדפיסה 22 (שני תווים) ומחזירה 2. לכן printf חיצונית (רביעית מימין) מדפיסה 2.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
     int z='z'-1;
     putchar(z-(int)pow((z^1)-z,z<<1));
}
```

יש לנמק את התשובה.

x

בשורה הראשונה משתנה z מקבל ערך שקטן ב-1 מקוד ASCII של 'z', ז"א קוד ASCII של 'y'. ייצוג בינארי של 1 מורכב מאפסים וסיבית 1 בסוף. לפי כללי פעולת ^ (XOR), a^0=a , a^1=a' . לכן z^1 שונה מ- z רק בסיבית ימנית ולכן הפרש ביניהם הוא 1 או 1-. אחרי הזזת שמאלה סיביות של z, הסיבית הימנית של z תהיה 0 ולכן זה מספר זוגי. לכן z^1 בחזקת z<<1 שווה ל- z .1 שווה לקוד ASCII של 'y' ואז z-1 שווה לקוד ASCII של 'x'. לכן הפלט הוא x.

שאלה 5 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
  printf("%d",strcmp("&","&&")&strcmp("&","&&")|("strcmp(&,&&)"&&"strcmp(&,&&)"));
}
```

יש לנמק את התשובה.

1-

המחרוזת "&" קטנה מילולית מהמחרוזת "&&". לכן ("&","&&")strcmp מחזירה 1-.
פעולת & (and) על סיביות שוות נותנת אותה סיבית. לכן 1- & 1- שווה ל- 1-.
מצד ימין מבצעים פעולה בוליאנית && על שתי המחרוזות, ז"א על כתובות שלהן אשר שונות מ- NULL. הפירוש הלוגי של כל מה ששונה מאפס הוא true ולכן ערך מספרי של תוצאת הפעולה הוא 1.
בסוף מבצעים את הפעולה 1 | 1- . ייצוג בינארי של 1- הוא 111...1 וייצוג בינארי של 1 הוא 000...001. פעולת | (or) על סיביות 1 ו- 0 וגם על סיביות 1 ו- 1 נותנת סיבית 1. לכן ייצוג בינארי של התוצאה הוא 111...1 , ז"א 1-.

שאלה 6 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

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

void main()
{
     puts("puts"-strlen("strlen")+sizeof"sizeof");
}
```

יש לנמק את התשובה.

<u>תשובה ונימוק:</u>

uts

פונקציה strlen מחזירה מספר תווים במחרוזת עד לתו אפסי לא כולל תו אפסי. לכן strlen("strlen") מחזירה 6.
אופרטור sizeof מחזיר את גודל האופרנד בבתים. מחרוזת "sizeof" מהווה 7 תווים כולל תו אפסי. לכן sizeof"sizeof" מחזיר 7.
ואז המשפט שקול ל- puts("puts"-6+7) ז"א ל- puts("puts"+1).
למחרוזת קבועה במשפט צריך להתייחס כמו לכתובת התו הראשון שלה. בזאת מקבלים למעשה puts("uts").
לכן הפלט יהיה uts .

שאלה 6 (בונוס – 10 נק')

מהו הפלט של התוכנית הבאה?

```
#include <stdio.h>

void main()
{
     char s[]={'^','^','^'^'^','^'};
     printf("%s",s);
}
```

יש לנמק את התשובה.

<u>תשובה:</u>

^^

<u>נימוק</u>

איבר מס' 0 ואיבר מס' 1 ב- s שווים ל- '^'. איבר מס' 2 הוא '^' ^ '^' , ז"א ערך של פעולת XOR על סיביות קודי ASCII של תו '^'. פעולת XOR על ערכים שווים נותנת 0. לכן הערך מורכב רק מסיביות 0, ז"א שווה ל- 0 אשר מהווה קוד ASCII של '\0'. פונקציה printf מדפיסה מחרוזת עד '\0' הראשון.
לכן יודפסו שני התווים הראשונים: ^^ .