

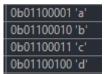
# אופרטורים בינאריים וריבוי תהליכים

### 1. ממשו את הפונקציה:

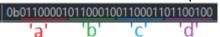
int pack4Chars2Int(char c1, char c2, char c3, char c4);

- a. הפונקציה מקבלת 4 תווים ומחזירה שלם שמכיל את ארבעת התווים.
- b של מנת לחבר את התווים bitwise OR על מנת להזיז את התוצאה ובפעולה shift על מנת לחבר את התווים. לתוצאה (לא בכפל בחזקה שלמה של 2 ובפעולת חיבור אריתמטית רגילה).

לדוגמא: עבור ארבעת התווים 'a', 'b', 'c', 'd', שייצוגם הבינארי הוא:



# הפונקציה תחזיר את השלם:



## 2. ממשו את הפונקציה:

int checkEvenParity(short input);

- הוא 2 בתים. אין צורך להוסיף קוד short ברוב המערכות ברוחב 2 בתים ברוחב 2 בתים שלם ברוחב 2 בתים שבודק זאת במטלה).
  - b. הפונקציה תחזיר 1 אם סכום הסיביות במספר זוגי, אחרת, הפונקציה תחזיר 0.

#### :לדוגמא

- עבור השלם: 14580- שבייצוג בינארי נראה כך: 0b1100011100001100, ובייצוג הקסאדצימאלי
   נראה כך: 0xC70C, הפונקציה תחזיר 0 כי מספר סיביות ה-1 הוא 7 (אי זוגי).
- עבור השלם 24, שבייצוג בינארי נראה כך: 0b000000000011000 , ובייצוג הקסאדצימאלי נראה כך: 0x0018, הפונקציה תחזיר 1, כי מספר סיביות ה-1 הוא 2 (זוגי).



3. נתון הקוד הבא:

```
1. #include <stdio.h>
2. #include <unistd.h>
3. #include <sys/wait.h>
4. #include <sys/types.h>
5.
6. main()
7. {
8.
           int status;
9.
           pid_t pid1, pid2;
10.
           pid_t wait_pidA, wait_pidB;
11.
           if ((pid1 = fork()) == 0)
12.
13.
                   printf("first child context\n");
14.
           else
15.
                   if ((pid2 = fork()) == 0)
16.
                          printf("second child context\n");
17.
18.
                   {
                          wait_pidA = wait(&status);
19.
20.
                          if (wait_pidA == pid1)
                                  printf("first child terminated\n");
21.
22.
                          if (wait_pidA == pid2)
                                  printf("second child terminated\n");
23.
24.
25.
                          wait_pidB = wait(&status);
26.
                          if (wait_pidB == pid1)
27.
                                  printf("first child terminated\n");
28.
                          if (wait_pidB == pid2)
29.
                                  printf("second child terminated\n");
30.
                   }
31.}
```

• רשמו 4 הדפסות אפשריות ו-2 הדפסות לא אפשריות.