1. **(20 نمره)**

الف- (10 نمره) آرایه‌ای از اعداد صحیح به طول n داریم که در آن اعدادی به صورت رندوم قرار دارند. شبه کدی بنویسید که عدد K را از کاربر دریافت کرده و چک کند که آیا در این ارایه دو عدد وجود دارند که جمع آن ها برابر K شود یا خیر.

مثال:

Array = [6,1,3,11,19,7,23,43]
k = 26 => true
k = 13 => false
K = 19 => false

ساده ترین حل این نوع مسایل بررسی تمام حالت های موجود هست که به این روش brute force میگوییم. به این شکل که تمام ترکیب های دوتایی داخل آرایه را با هم جمع بزنیم و بررسی کنیم که حاصل برابر k می باشد یا خیر.

```
for i from 0 to n do
    for j from i to n do
        if Array[i] + Array[j] = k then
            return true
        endif
    endfor
endfor
return false
```

ب- (**10 نمره**) تابعی به نام checkSum بنویسید که آرایه، طول آن n و عدد K را به عنوان آرگومان دریافت کرده و شبه کد بند الف را اجرا کند. مقدار برگشتی این تابع 1 برای حالت TRUE (دو عدد که جمعشان مساوی K شود در آرایه وجود دارد) و 0 برای حالت FALSE (دو عدد که جمعشان مساوی K شود در آرایه وجود ندارد)، خواهد بود.

```
int checkSum (int Array [], int n, int k){
    for (int i = 0; i < n; i ++){
        for (int j = i + 1; j < n; j ++){
            if (Array[i] + Array[j] == k){
                return 1;
            }
        }
    }
    return 0;
}
```

2. (**20 نمره**) در بازی minesweeper یک جدول n * n داریم که در تعدادی از آنها بمب قرار دارد. همچنین امتیاز یک خانه برابر است با تعداد بمب‌هایی که در 9 خانه مجاور آن قرار دارد. ما زمین بازی را با یک آرایه دو بعدی از int پیاده سازی کردیم به صورتی که اگر مقدار field[i][j] 1 مساوی باشد به این معناست که در خانه i و j بمب قرار دارد و اگر 0 باشد یعنی بمبی در آن خانه وجود ندارد. با فرض اینکه ابعاد جدول ما n=8 می باشد، تابعی بنویسید که در ورودی آرایه دو بعدی field و مختصات i و j را در ورودی دریافت کند و امتیاز آن خانه را در خروجی برگرداند.

نکته ای که در این سوال باید به آن دقت شود، مرز های جدول است.

اگر یکی از i یا j در مرز قرار داشته باشند، خانه های مجاور تعدادشان 5 تا می شود که باید نتیجه از حاصل جمع آن ها بدست می آید.

اگرهر دوی i و j در مرز باشند، یعنی خانه ی iوj در یکی از چهار گوشه ی جدول باشد، تعداد خانه های مجاور برابر 3 می شود.

و در آخر هم اگر هیچ کدام مرزی نباشند، تعداد خانه های مجاور که باید با هم جمع زده شوند برابر 8 می باشد.

```
int CalculatePointOfCell(int field[8][8], int i, int j){
    int result = field[i][j];

    switch(i){
        case 0:
            if (j == 0){
                result = field[0][1] + field[1][0] + field[1][1];
            }
            else if (j == 7)
            {
                result = field[0][6] + field[1][7] + field[1][6];
            }
            else{
                result = field[0][j-1] + field[1][j-1] + field[1][j] + field[1][j+1] + field[0][j+1];
            }
            break;

        case 7:
            if (j == 0){
                result = field[6][0] + field[7][1] + field[6][1];
            }
            else if (j == 7)
            {
                result = field[7][6] + field[6][7] + field[6][6];
            }
            else{
                result = field[7][j-1] + field[6][j-1] + field[6][j] + field[6][j+1] + field[7][j+1];
            }
            break;

        default:
            if (j == 0){
                result = field[i-1][0] + field[i-1][1] + field[i][1] + field[i+1][1] + field[i+1][0];
            }
            else if (j == 7)
            {
                result = field[i-1][7] + field[i-1][6] + field[i][6] + field[i+1][6] + field[i+1][7];
            }
            else{
                result = field[i-1][j-1] + field[i][j-1] + field[i+1][j-1] + field[i+1][j]
                       + field[i+1][j+1] + field[i][j+1] + field[i-1][j+1] + field[i-1][j];
            }
    }
}
```

3. (**20 نمره**) یک جدول دو بعدی به ابعاد 3*4 از اعداد int داریم. تابعی بنویسید که در ورودی آرایه دو بعدی Table و دو مقدار int به نام‌های rotateLeft و rotateUp دریافت کند و جدول را ابتدا به تعداد rotateUp به سمت بالا چرخش دهد و سپس به تعداد rotateLeft به سمت چپ چرخش دهد.
منظور از چرخش این است که وقتی عددی مثلا از بالا (و یا سمت چپ) این جدول دو بعدی خارج می شود از سمت پایین (و یا راست) وارد می شود.

```
void RotateTable(int Table[4][3], int rotateUp ,int rotateLeft){
}
```

مثال:

Table:

2	14	19
5	3	8
9	21	6
11	61	13

```
RotateTable(Table[4][3], 1, 2); // 2 left, 1 up
```

Result:

8	5	3
6	9	21
13	11	61
19	2	14

```
72  void RotateTable(int Table[4][3], int rotateUp, int rotateLeft) {
73      // Rotate rows upward
74      for (int i = 0; i < rotateUp; i++) {
75          // Save the first row
76          int temp[3];
77          for (int j = 0; j < 3; j++) {
78              temp[j] = Table[0][j];
79          }
80
81          // Shift rows up
82          for (int r = 0; r < 3; r++) { // Use 3, as the last row will be replaced
83              for (int c = 0; c < 3; c++) {
84                  Table[r][c] = Table[r + 1][c];
85              }
86          }
87
88          // Copy temp to the last row
89          for (int j = 0; j < 3; j++) {
90              Table[3][j] = temp[j];
91          }
92      }
```

```
94      // Rotate columns left
95      for (int i = 0; i < rotateLeft; i++) {
96          // Save the first column
97          int temp[4];
98          for (int j = 0; j < 4; j++) {
99              temp[j] = Table[j][0];
100         }
101
102         // Shift columns left
103         for (int c = 0; c < 2; c++) { // Use 2, as the last row will be replaced
104             for (int r = 0; r < 4; r++) {
105                 Table[r][c] = Table[r][c + 1];
106             }
107         }
108
109         // Copy temp to the last column
110         for (int j = 0; j < 4; j++) {
111             Table[j][2] = temp[j];
112         }
113     }
114 }
```

4. (**10 نمره**) شبه کد الگوریتمی را بنویسید که آرایه‌ای به طول n از اعداد صحیح را دریافت کرده و تعداد تکرار بزرگ ترین عدد داخل آرایه را در خروجی چاپ نماید.

ابتدا دو متغیر با نام های count و max تعریف میکنیم. Max را برابر اولین عدد داخل آرایه میگذاریم و count را 1 میکنیم. سپس به ترتیب در آرایه جلو میرویم و در هر خانه چک میکنیم که آیا عدد با max برابر است یا خیر، اگر برابر بود به count یک اضافه میکنیم. اگر نبود و عدد بزرگ تر از max بود، آنگاه max را برابر عدد میکنیم و همچنین count را مساوی 1 میگذاریم. پس همواره max بزرگترین عدد دیده شده در آرایه، و همچنین count برابر تعداد دیده شدن های max است.

```
initialize count to 1
initialize max to Array[0]
for i from 1 to length(Array) do
    if Array[i] > max then
        max = Array[i]
        count = 1
        continue
    endif
    if Array[i] = max then
      count += 1
    endif
endfor
```

5. **سوال امتیازی ** (20 نمره)**

رشته‌ای از کاراکتر داریم که در هر خانه آن "(" یا ")" قرار دارد. شبه کدی بنویسید که ارزش عددی این رشته را با توجه به سه قانون زیر بدست آورد:

قانون یک: مقدار () برابر است با یک.

قانون دو: مقدار (()) برابر است با 2 به توان پرانتز داخلی.

قانون سوم: مقدار ()() برابر است با جمع دو مقدار پرانتز ها.

مثال:

() = 1
(()) = 2
((())) = 4
()() = 1
(())() = 3
()(()()()()()) = 18
((()))(()())) = 256

- **نکته:** تضمین می‌شود که فرمت پرانتزها صحیح و تعداد پرانتزهای باز و بسته مساوی است. یعنی هر پرانتز ها حتما بسته می‌شود و همچنین امکان ندارد پرانتز بسته بعد از پرانتز باز متناظر خود بیاید.

یکی از راه های حل این مسئله استفاده از روش بازگشتی است. بدین صورت که جواب هر پرانتز را برابر 2 به توان حاصل داخل پرانتز قرار می دهیم و همچنین چک میکنیم اگر عبارت تنها یک () بود مقدار 1 را بازگرداند. اما حالت دیگری که باید آن هم در نظر بگیریم وقتی است که قرار است تو پرانتز با هم جمع شوند، پس باید قبل از بتوان رساندن چک کنیم که اگر پرانتزی نمانده که بسته نشده باشد، ولی هم چنان به آخر متن نرسیده ایم، بیایم و سمت چپ و راست را جداگانه حساب کرده و با هم جمع بزنیم.

```
Function get_score(string):
    If string is empty:
        Return 0
    EndIf

    If length of string is 2:
        Return 1
    EndIf

    Initialize delta to 0
    Initialize len to length(string)

    For i from 0 to len do:

        If delta = 0 and i != 0 and i != length - 1:
            Return get_score(substring from index 0 to i - 1) + get_score(substring from i to end)
        EndIf

        If string[i] = '(':
            delta += 1
        EndIf

        If string[i] = ')':
            delta -= 1
        EndIf
    EndFor

    Return 2 ** get_score(substring from index 1 to i - 1)

EndFunction
```

کد با زبان c آن نیز به شکل زیر می شود:

```c
int get_score(char* string) {
    int len = strlen(string);

    // در صورت خالی بودن 0 برگرداند
    if (len == 0) {
        return 0;
    }

    //اگر () بود 1 برگرداند
    if (len == 2) {
        return 1;
    }

    int delta = 0;

    for (int i = 0; i < len; i++) {

        if (delta == 0 && i != 0 && i != len - 1) { // چک کردن نیاز به شکاندن آرایه به دو قسمت
            //                                      برای وقتی که باید دو قسمت را جمع بزنیم

            char left[100], right[100];
            strncpy(left, string, i);      // کپی کردن قسمت سمت چپ و راست
            left[i] = '\0';
            strcpy(right, string + i);
            return get_score(left) + get_score(right);
        }

        if (string[i] == '(') {
            delta++;
        } else if (string[i] == ')') {
            delta--;
        }
    }

    // اگر نیازی به جمع زدن نباشد، یعنی پرانتز ها داخل هم هستند و هیچ دو پرانتزی کنار هم نیستند
    char sub[100];
    strncpy(sub, string + 1, len - 2);
    sub[len - 2] = '\0';
    return get_score(sub);
}
```