```
//question 1 part A
n=INPUT INT
array = INPUT ARRAY with size n
k = INPUT INT
i=0
j=0
flag=0
FOR i to n-1 with steps +1
        IF flag=1 THEN
        break
        ENDIF

        FOR j to n-1 with steps +1
        IF i=j THEN
        continue
        ENDIF
        IF array[i]+array[j]=k THEN
        flag=1
        break
        ENDIF
        END FOR
END FOR
IF flag=1 THEN
        print true
ELSE
        print false
ENDIF
```

```
// question 1 part B

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

```c
//question 2
#include <stdio.h>
int CalculatePointOfCell (int field[8][8], int i, int j)
{
   int new_field[10][10];//padding 0 around of field array and
then calculate sum of neighbor
   for(int i_1=0 ; i_1<10 ; i_1++)
   {
       for(int j_1=0 ; j_1<10 ;j_1++)
       {
           if(i_1==0||i_1==9)
           {
               new_field[i_1][j_1]=0;
               continue;
           }
           if(j_1==0 || j_1==9)
           {
               new_field[i_1][j_1]=0;
           }
           else
           {
               new_field[i_1][j_1]=field[i_1-1][j_1-1];
           }
       }
   }
   i=i+1 ;
   j=j+1 ;
   int answer =0 ;
   answer = new_field[i][j] + new_field[i][j+1] +
new_field[i][j-1]+new_field[i+1][j]+new_field[i-1][j]+new_field[
i+1][j+1]+new_field[i-1][j+1]+new_field[i-1][j-1]+new_field[i+1]
[j-1];
   return answer  ;

}
```

```
//question 3
void RotateTable(int Table[4][3], int rotateUp, int rotateLeft)
{
   int temp;
   // i change value of rotate by % to prevent useless loop
   int rows=4 , columns=3 ;
   rotateUp=rotateUp%4 ;
   rotateLeft=rotateLeft%3;
   for (int i = 0; i < rotateUp; i+=1)
    {
       for (int j = 0; j < columns; j+=1)
       {
           temp = Table[0][j];
           for (int k = 0; k < columns; k+=1)
           {
               Table[k][j] = Table[k + 1][j];
           }
           Table[3][j] = temp;
       }
   }

   for (int i = 0; i < rotateLeft; i+=1)
   {
       for (int j = 0; j < rows; j+=1)
       {
           temp = Table[j][0];
           for (int k = 0; k < columns-1; k+=1)
           {
               Table[j][k] = Table[j][k + 1];
           }
           Table[j][2] = temp;
       }
   }
}
```

```
//question 4
n=INPUT INT
array = INPUT INT ARRAY with size n
biggest=array[0]
i=0
FOR i to n-1 with steps +1
        IF array[i]>biggest THEN
        biggest=array[i]
        ENDIF
ENDFOR
answer=0
i=0
FOR i to n-1 with steps +1
        IF array[i]=biggest THEN
        answer+=1
        ENDIF
ENDFOR
print answer
```

```
//question 5
string=INPUT CHAR ARRAY
answer=0
index=0
// i assume in pseudocode function , function change the original input
FUNCTION calculator (CHAR ARRAY string)
        WHILE string[index]!='\0'
        IF string[index]='(' THEN
        index+=1
        IF string[index]=')' THEN
                answer+=1
                index+=1
        ELSE
                next_parentheses=calculator(string)
                answer = answer+(2^next_parentheses)
                 // i assume ^ means power like 2^3=8
        ENDIF
        ELSE IF string[index]=')'  THEN
                index+=1
                Break
        ELSE
                index+=1
        ENDIF
        END WHILE
        return answer
END FUNCTION
final_answer=calculator(string)
print final_answer
```