2. (b) CTS Test 2

Test Summary

- No. of Sections: 1
- No. of Questions: 7
- Total Duration: 20 min

Section 1 - Automata

Section Summary

- No. of Questions: 7
- Duration: 20 min

Additional Instructions:

None

Q1. **Find the factorial of a given number.**

Sample Input

```
5
```

Sample Output

```
120
```

Time Limit: 2 ms Memory Limit: 256 kb Code Size: 256 kb

Q2. **Print the prime numbers from an array up to given value n by using existing function.**

Sample Input

```
10
```

Sample Output

```
2 3 5 7
```

Time Limit: 2 ms Memory Limit: 256 kb Code Size: 256 kb

Q3. Generate all the prime numbers below a given **N**

Sample Input

```
30
```

Sample Output

```
2 3 5 7 11 13 17 19 23 29
```

Time Limit: 2 ms Memory Limit: 256 kb Code Size: 256 kb

Q4. The function **checkGrade(int marks)** is supposed to return a student's grade when the student's test marks is passed to it as an argument (0 <= marks <= 100)
Given a particular marks, a grade is calculated as per the following table:

Score	Grade
1. marks>=91	**A**
2. 76<=marks<=90	**B**
3. 61<=marks<=75	**C**
4. marks<=60	**D**

Sample Input

```
91
```

Sample Output

```
A
```

Time Limit: 2 ms Memory Limit: 256 kb Code Size: 256 kb

Q5. Encode the given string.
For Example:
Input:

aaaabbbccdefaa

Output:

a4b3c2d1e1f1a2

Sample Input

```
aaaabbbccdefaa
```

Sample Output

```
a4b3c2d1e1f1a2
```

Time Limit: 2 ms Memory Limit: 256 kb Code Size: 256 kb

Q6. A integer Array is given . You need to find the maximum sum contiguous subset.

Sample Input

```
6
5 6 0 -6 5 -8
```

Sample Output

```
11
```

Time Limit: 2 ms Memory Limit: 256 kb Code Size: 256 kb

Q7. print the spiral of the matrix

Sample Input

```
4
```

Sample Output

```
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
```

Time Limit: 2 ms Memory Limit: 256 kb Code Size: 256 kb

Answer Key & Solution

Section 1 - Automata

Q1

Test Case

Input	Output
10	3628800

Weightage - 50

Input	Output
4	24

Weightage - 50

Sample Input	Sample Output
5	120

Solution

Header

```
#include<stdio.h>
int factorial(int n)
{
```

```
#include<stdio.h>
int factorial(int n)
{
    int itr,fact=1;
for(itr =1; itr <= n; itr++)
{
fact = fact * itr;
}
return fact;
}
int main()
{
long int fact, n, i;
scanf("%d", &n);
fact=factorial(n);
printf("%d", fact);
return 0;
}
```

Footer

```
}
int main()
{
int fact, n, i;
scanf("%d", &n);
fact=factorial(n);
printf("%d", fact);
return 0;
}
```

Q2

Test Case

Input

```
26
```

Output

```
2 3 5 7 11 13 17 19 23
```

Weightage - 20

Input

```
150
```

Output

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59
```

Weightage - 50

Input

```
10
```

Output

```
2 3 5 7
```

Weightage - 30

Sample Input

```
10
```

Sample Output

```
2 3 5 7
```

Solution

Header

```
#include<stdio.h>
int isprime(int num)
{
```

```
#include<stdio.h>
int isprime(int num)
{
 int i;

  int isprime = 1;
  for(i = 2;  i <= num / 2; i++)
```

```
  {
          if(num % i == 0)
          {
                  isprime = 0;
                       break;
          }
   }
   return isprime;
}
int main()
{
int n, m, arr[100], size=0, i;
scanf("%d", &n);
for(m = 2; m <= n; m++)
{
if(isprime(m))
    arr[size++]= m;
}
for(i = 0; i < size; i++)
{
printf("%d ", arr[i]);
}
return 0;
}
```

Footer

```
}
int main()
{
int n, m, arr[100], size=0, i;
scanf("%d", &n);
for(m = 2; m <= n; m++)
{
if(isprime(m))
    arr[size++]= m;
}
for(i = 0; i < size; i++)
{
printf("%d ", arr[i]);
}
return 0;
}
```

Q3

Test Case

Input	Output
100	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59

Weightage - 25

Input	Output
100000	2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59

Weightage - 50

Input

```
1000000
```

Output

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59
```

Weightage - 25

Sample Input

```
30
```

Sample Output

```
2 3 5 7 11 13 17 19 23 29
```

Solution

Header

```
#include<stdio.h>
#include<math.h>
int main(){
    int n;
    scanf("%d",&n);
```

```
#include<stdio.h>
#include<math.h>
int main(){
    int n;
    scanf("%d",&n);
    int arr[n-1];
    int sqr=sqrt(n)+1;
    for(int i=0;i<n-1;i++){
        arr[i]=i+2;
    }
  for(int i=0;arr[i]<=sqr;i++){
      if(arr[i]==0){
          continue;
      }
      for(int j=i+arr[i];j<n-1;j+=(arr[i])){
          arr[j]=0;
      }
  }
for(int i=0;i<n-1;i++){
    if(arr[i]!=0){
        printf("%d ",arr[i]);
    }
}
}
```

Footer

```
}
```

Q4

Test Case

Input

76

Output

B

Weightage - 25

Input

75

Output

C

Weightage - 25

Input

61

Output

C

Weightage - 25

Input

60

Output

D

Weightage - 25

Sample Input

91

Sample Output

A

Solution

Header

```
#include<stdio.h>
void checkGrades(int marks){



#include<stdio.h>
void checkGrades(int marks){
    if(marks>=91){
        printf("A");
    }
    else if(marks>=76 && marks<=90){
        printf("B");
    }
    else if(marks>=61 && marks<=75){
        printf("C");
    }
    else{
        printf("D");
    }
```

```
    }


}
int main()
{
    int marks;
    scanf("%d",&marks);
    checkGrades(marks);
}
```

Footer

```
}
int main()
{
    int marks;
    scanf("%d",&marks);
    checkGrades(marks);
}
```

Q5

Test Case

Input

```
kjkfhskgkbvkjsduivgsd
```

Output

```
k1j1k1f1h1s1k1g1k1b1v1k1j1s1d1u1i1v1g1s1d1
```

Weightage - 25

Input

```
jcjcjcjjcjcjcjcjjcj
```

Output

```
j1c1j1c1j1c1j2c1j1c1j1c1j1c1j2c1j1
```

Weightage - 25

Input

```
hjhhjhhhhhhhhhhh
```

Output

```
h1j1h2j1h12
```

Weightage - 50

Sample Input

```
aaaabbbccdefaa
```

Sample Output

```
a4b3c2d1e1f1a2
```

Solution

Header

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

```
#include<math.h>
int main(){
    char arr[1000];
    scanf("%s",arr);
    char alpha[1000];
    int num[1000];
    int front=-1,rear=-1;
    int len=strlen(arr);
    alpha[++rear]=arr[0];
    num[rear]=1;



#include<stdio.h>
#include<string.h>
#include<math.h>
int main(){
    char arr[1000];
    scanf("%s",arr);
    char alpha[1000];
    int num[1000];
    int front=-1,rear=-1;
    int len=strlen(arr);
    alpha[++rear]=arr[0];
    num[rear]=1;
    for(int i=1;i<len;i++){
        if(arr[i]==alpha[rear]){
            num[rear]++;
        }
        else{
            alpha[++rear]=arr[i];
            num[rear]=1;
        }
    }
    while(front!=rear){
        ++front;
        printf("%c%d",alpha[front],num[front]);
    }

}
```

Footer

```
    while(front!=rear){
        ++front;
        printf("%c%d",alpha[front],num[front]);
    }

}
```

Q6

Test Case

Input	Output
5 1 9 -6 -7 8 9	10

Weightage - 50

Input	Output

```
10
-7 -6 -4 4 8 9 -4 5 -9
```

```
22
```

Weightage - 50

Sample Input

```
6
5 6 0 -6 5 -8
```

Sample Output

```
11
```

Solution

Header

```
#include<stdio.h>
int maxi(int arr[],int n){
```

```
#include<stdio.h>
int flag=-1;
int max(int a,int b){
    if(a>=b){
        flag=1;
        return a;
    }
    else{
        flag=0;
        return b;
    }
}
int main()
{
    int n,k;
    scanf("%d",&n);
    int arr[n],dummy[n],m=0;
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    int g=0,l=0;
    for(int i=0;i<n;i++){
        l=max(arr[i]+l,arr[i]);
        if(flag==1){
            dummy[m++]=arr[i];
        }
        else if(flag==0 && l>g){
            m=0;
            dummy[m++]=arr[i];
        }
        if(l>g){
            g=l;
        }
    }
    printf("%d",g);


}
```

Footer

```
}
int main()
{
    int n,k;
    scanf("%d",&n);
    int arr[n],dummy[n],m=0;
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }

 int ans = maxi(arr,n);
 printf("%d",ans);

}
```

Q7

Test Case

Input

```
5
```

Output

```
5 5 5 5 5 5 5 5 5
5 4 4 4 4 4 4 4 5
5 4 3 3 3 3 3 4 5
5 4 3 2 2 2 3 4 5
```

Weightage - 50

Input

```
7
```

Output

```
7 7 7 7 7 7 7 7 7 7 7 7 7
7 6 6 6 6 6 6 6 6 6 6 6 7
7 6 5 5 5 5 5 5 5 5 5 6 7
7 6 5 4 4 4 4 4 4 4 5 6 7
```

Weightage - 50

Sample Input

```
4
```

Sample Output

```
4 4 4 4 4 4 4
4 3 3 3 3 3 4
4 3 2 2 2 3 4
4 3 2 1 2 3 4
```

Solution

Header

```
#include<stdio.h>
#include<string.h>
#include<math.h>
void pattern(int n){
```

```
#include<stdio.h>
#include<string.h>
#include<math.h>
int main()
{
   int n;
   scanf("%d",&n);
   int size=2*n-1;
   int arr[size][size];
```

```
int u=0,l=0,b=size,r=size,val=n;
 while(l<r && u<b){
     for(int i=l;i<r;i++){
         arr[u][i]=val;
     }
     u++;
     for(int i=u;i<b;i++){
         arr[i][r-1]=val;
     }
     r--;
     for(int i=r-1;i>=l;i--){
         arr[b-1][i]=val;
     }
     b--;
     for(int i=b-1;i>=u;i--){
         arr[i][l]=val;
     }
     l++;
     val--;
 }
 for(int i=0;i<size;i++){
     for(int j=0;j<size;j++){
         printf("%d ",arr[i][j]);
        }
        printf("\n");
    }
}
```

Footer

```
}
int main()
{

  int n;
   scanf("%d",&n);
   pattern(n);
}
```