

LAB 1**CSE5001 - Algorithm Design and Implementation**

1)

```

#include <stdio.h>
int main()
{
    // Question 1
    // 20MAI1001 PRABHU

    int sum,i;
    int x[] = {4,5,2,5,6,4,7};
    int n = sizeof(x)/sizeof(x[0]);
    //sum=0;
    for( i=0; i<=n-2; i++) {
        sum += ( (x[i]+x[i+1]) * x[i+2]);
        printf("(%d + %d)*%d - ",x[i],x[i+1],x[i+2]);
        printf("%d \n",sum);
    }
    printf("\n= %d",sum);

    return 0;
}

```

The screenshot shows a web browser with multiple tabs, including 'Database Courses', 'Introduction to Te...', 'Face Mask Detect...', 'Untitled5.ipynb', 'Database System', 'DS4_A01.qxd', 'notepad.pw / pgs...', and 'Online C Compiler'. The active tab is 'Online C Compiler'. The browser address bar shows 'tutorialspoint.com/compile_c_online.php'. The page title is 'Compile and Execute C Online (GNU GCC v7.1.1)'. The interface includes a 'Fork' button, a 'Project' dropdown, an 'Edit' button, a 'Setting' dropdown, and a 'Login' button. The code editor shows the following C code:

```

1 // Online C compiler to run C online.
2 // Write C code in this online editor and run it.
3
4 #include <stdio.h>
5
6
7 int main()
8 {
9     // Question 8
10    // 20MAI1001 PRABHU
11
12    int sum,i;
13    int x[] = {4,5,2,5,6,4,7};
14    int n = sizeof(x)/sizeof(x[0]);
15    //sum=0;
16    for( i=0; i<=n-2; i++) {
17        sum += ( (x[i]+x[i+1]) * x[i+2]);
18        printf("(%d + %d)*%d - ",x[i],x[i+1],x[i+2]);
19        printf("%d \n",sum);
20    }
21    printf("\n= %d",sum);
22
23    return 0;
24 }
25
26

```

The 'Result' tab shows the output of the program:

```

$gcc -o main *.c
$main
(4 + 5)*2 - 18
(5 + 2)*5 - 53
(2 + 5)*6 - 95
(5 + 6)*4 - 139
(6 + 4)*7 - 209
(4 + 7)*0 - 209
= 209

```

2)

// Online C compiler to run C online.

// Write C code in this online editor and run it.

#include <stdio.h>

int main()

{

// Question 2

// 20MAI1001 PRABHU

int sum,i;

int x[] = {4,5,2,5,6,4,7};

int n = sizeof(x)/sizeof(x[0]);

sum=1;

for(i=0; i<=n-3; i++) {

sum *= (x[i]+x[i+2]);

printf("(%d + %d) - ",x[i],x[i+2]);

printf("%d \n",sum);

}

printf("\n= %d",sum);

return 0;

}

The screenshot shows a web browser window with the URL `tutorialspoint.com/compile_c_online.php`. The page title is "Compile and Execute C Online (GNU GCC v7.1.1)". The code editor contains the following C code:

```

1 // Online C compiler to run C online.
2 // Write C code in this online editor and run it.
3
4 #include <stdio.h>
5
6
7 int main()
8 {
9     // Question 2
10    // 20MAI1001 PRABHU
11
12    int sum,i;
13    int x[] = {4,5,2,5,6,4,7};
14    int n = sizeof(x)/sizeof(x[0]);
15    sum=1;
16    for( i=0; i<=n-3; i++) {
17        sum *= (x[i]+x[i+2]);
18        printf("(%d + %d) - ",x[i],x[i+2]);
19        printf("%d \n",sum);
20    }
21    printf("\n= %d",sum);
22
23    return 0;
24 }
25
26

```

The output window shows the result of the compilation and execution:

```

$gcc -o main *.c

$main

(4 + 2) - 6
(5 + 5) - 60
(2 + 6) - 480
(5 + 4) - 4320
(6 + 7) - 56160
= 56160

```

3)

// Online C compiler to run C online.

// Write C code in this online editor and run it.

#include <stdio.h>

int main()

{

// Question 3

// 20MAI1001 PRABHU

int sum,i;

int x[] = {4,5,2,5,6,4,7};

int n = sizeof(x)/sizeof(x[0]);

sum=0;

for(i=0; i<=n-3; i++) {

sum += (x[i]-x[i+1]) * (x[i+1]+x[i+2]);

printf("(%d - %d) * (%d + %d)",x[i],x[i+1],x[i+1],x[i+2]);

printf("%d \n",sum);

}

printf("\n= %d",sum);

return 0;

}

The screenshot shows a web browser with multiple tabs. The active tab is 'Online C Compiler'. The browser address bar shows 'tutorialspoint.com/compile_c_online.php'. The compiler interface has a header bar with 'codingground' logo and 'Compile and Execute C Online (GNU GCC v7.1.1)'. Below the header, there are tabs for 'Execute', 'Share', 'main.c', and 'STDIN'. The 'main.c' tab is active, displaying the C code. The code is as follows:

```

1 // Online C compiler to run C online.
2 // Write C code in this online editor and run it.
3
4 #include <stdio.h>
5
6
7 int main()
8 {
9     // Question 3
10    // 20MAI1001 PRABHU
11
12    int sum,i;
13    int x[] = {4,5,2,5,6,4,7};
14    int n = sizeof(x)/sizeof(x[0]);
15    sum=0;
16    for( i=0; i<=n-3; i++) {
17        sum += (x[i]-x[i+1]) * (x[i+1]+x[i+2]);
18        printf("(%d - %d) * (%d + %d)",x[i],x[i+1],x[i+1],x[i+2]);
19        printf("%d \n",sum);
20    }
21    printf("\n= %d",sum);
22
23    return 0;
24 }
25
26

```

On the right side of the interface, there is a 'Result' section. It shows the command '\$gcc -o main *.c' and the output of the program:

```

$main
(4 - 5) * (5 + 2)-7
(5 - 2) * (2 + 5)14
(2 - 5) * (5 + 6)-19
(5 - 6) * (6 + 4)-29
(6 - 4) * (4 + 7)-7
= -7

```

4)

// Online C compiler to run C online.

// Write C code in this online editor and run it.

```
#include <stdio.h>
int main() {
    // Question 4
    // 20MAI1001 PRABHU
    int i;
    int data[] = {1,6,9,4,2,3,4,6,4,2,4,3,4,8,4,9};
    int n = sizeof(arr1)/sizeof(arr1[0]);
    printf("List of all Local Maxima \n\n");

    for(i=1;i<n-1;i++){
        if( data[i-1] < data[i] && data[i+1] < data[i]){
            printf("%d -- Left [%d] Right[%d]\n",data[i],data[i-1],data[i+1]);
        }
    }
    return 0;
}
```

The screenshot shows the Programiz Online C Compiler interface. The code editor on the left contains the C program, and the output window on the right displays the results of the compilation and execution.

Code Editor:

```
main.c
1 // Online C compiler to run C online.
2 // Write C code in this online editor and run it.
3
4 #include <stdio.h>
5 int main() {
6     // Question 4
7     // 20MAI1001 PRABHU
8     int i;
9     int data[] = {1,6,9,4,2,3,4,6,4,2,4,3,4,8,4,9};
10    int n = sizeof(data)/4;
11    printf("List of all Local Maxima \n\n");
12
13    for(i=1;i<n-1;i++){
14        if( data[i-1] < data[i] && data[i+1] < data[i]){
15            printf("%d -- Left [%d] Right[%d]\n",data[i],data[i-1],
16                data[i+1]);
17        }
18    }
19    return 0;
20 }
```

Output:

```
gcc -o /tmp/Gt5pWhxxTL.o /tmp/Gt5pWhxxTL.c -lm
/tmp/Gt5pWhxxTL.o
List of all Local Maxima
9 -- Left [6] Right[4]
6 -- Left [4] Right[4]
4 -- Left [2] Right[3]
8 -- Left [4] Right[4]
```

5)

// Online C compiler to run C online.

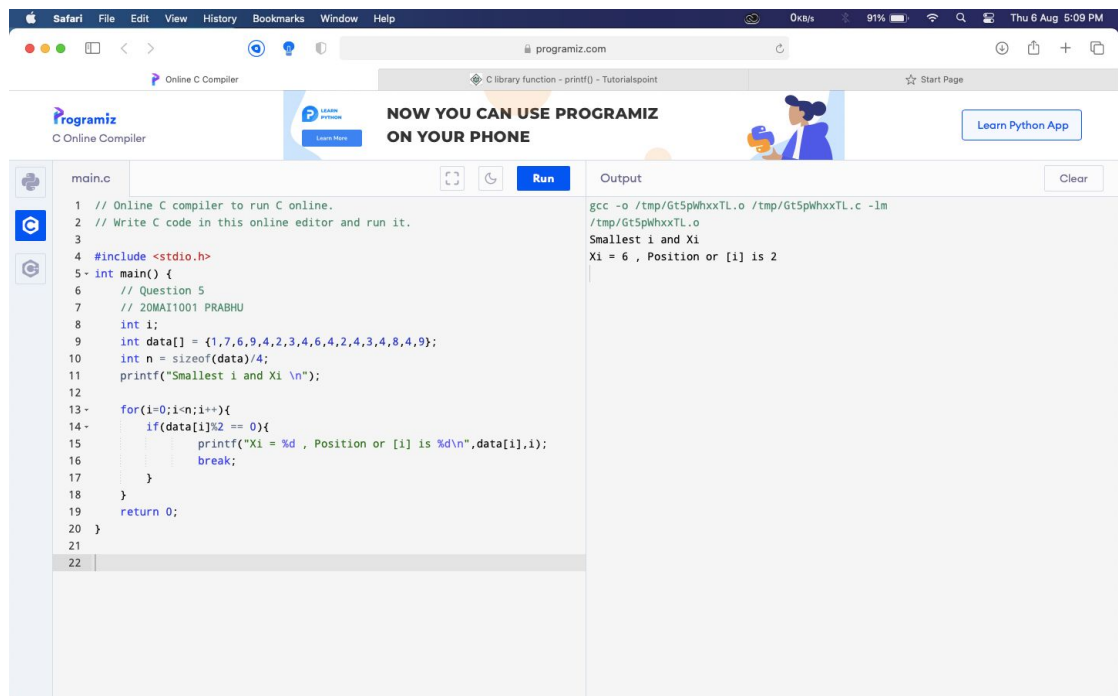
// Write C code in this online editor and run it.

```

#include <stdio.h>
int main() {
    // Question 5
    // 20MAI1001 PRABHU
    int i;
    int data[] = {1,7,6,9,4,2,3,4,6,4,2,4,3,4,8,4,9};
    int n = sizeof(arr1)/sizeof(arr1[0]);
    printf("Smallest i and Xi \n");

    for(i=0;i<n;i++){
        if(data[i]%2 == 0){
            printf("Xi = %d , Position or [i] is %d\n",data[i],i);
            break;
        }
    }
    return 0;
}

```



6)

// Online C compiler to run C online.

// Write C code in this online editor and run it.

#include <stdio.h>

int main() {

// Question 6

// 20MAI1001 PRABHU

int i;

int data[] = {1,7,6,9,4,2,3,4,6,4,2,4,3,4,8,4,9};

int n = sizeof(data)/4;

printf("Smallest i where consecutive are even \n");

for(i=0;i<n-1;i++){

if(data[i]%2 == 0 && data[i+1]%2 == 0){

printf("Position or [i] is %d -- the numbers are %d and

%d\n",i,data[i],data[i+1]);

break;

}

}

return 0;

}

The screenshot shows the Programiz Online C Compiler interface. The code editor on the left contains the C program, and the output window on the right shows the execution results.

```

main.c
1 // Online C compiler to run C online.
2 // Write C code in this online editor and run it.
3
4 #include <stdio.h>
5 int main() {
6     // Question 6
7     // 20MAI1001 PRABHU
8     int i;
9     int data[] = {1,7,6,9,4,2,3,4,6,4,2,4,3,4,8,4,9};
10    int n = sizeof(data)/4;
11    printf("Smallest i where consecutive are even \n");
12
13    for(i=0;i<n-1;i++){
14        if(data[i]%2 == 0 && data[i+1]%2 == 0){
15            printf("Position or [i] is %d -- the numbers are %d and
16                %d\n",i,data[i],data[i+1]);
17            break;
18        }
19    }
20    return 0;
21 }
22

```

Output

```

gcc -o /tmp/Gt5pWhxxTL.o /tmp/Gt5pWhxxTL.c -lm
/tmp/Gt5pWhxxTL.o
Smallest i where consecutive are even
Position or [i] is 4 -- the numbers are 4 and 2

```

7)

// Online C compiler to run C online.

// Write C code in this online editor and run it.

#include <stdio.h>

int* swap(int *arr,int pos)

```
{
    int head = arr[0];
    arr[0] = arr[pos];
    arr[pos] = head;
    return arr;
}
```

void printIt(int *arr,int n){

```
    for (int c = 0; c < n ; c++)
        printf("%d ", arr[c]);
}
```

int* crop(int* arr){

```
    arr = arr;
    return arr;
}
```

int main()

{

// Question 7

// 20MAI1001 PRABHU

int newArr[100] = { 0 };

int arr1[]={8,53,87,23,6,3,10,20,30,40};

int* arr = arr1;

int n = sizeof(arr1)/sizeof(arr1[0]);

int newPos = 0;

int dup_n = n;

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

int head = *arr;

printf("%d'th iteration - ",i);

printf("dup_n = %d\n",dup_n);

for (int j = 1; j < dup_n; j++) {

if (*(arr+j) < head){

//printf("%d - %d - %d\n",head,*(arr+j),dup_n);

head = *(arr+j);

swap(arr,j);

```

        //printf(arr,dup_n);
    }
}
newArr[newPos] = head;
newPos++;
head = 0;
dup_n--;
//crop(arr);
printf("Input array - ");
printf(arr,dup_n);
arr++;
printf("\n");
printf("New array - ");
printf(newArr,n-dup_n);
printf("\n\n");
}

printf("Final non-decreasing array - ");
printf(newArr,n);
return 0;
}

```

The screenshot shows the CodingGround online C compiler interface. The code is a C program that sorts an array using a stack-based algorithm. The output shows the state of the array after each iteration of the sorting process.

```

1 // Online C compiler to run C online.
2 // Write C code in this online editor and run it.
3
4 #include <stdio.h>
5 int* swap(int *arr,int pos)
6 {
7     /* Some operations on arr[] */
8     int head = arr[0];
9     arr[0] = arr[pos];
10    arr[pos] = head;
11    return arr;
12 }
13
14 void printIt(int *arr,int n){
15     for (int c = 0; c < n; c++){
16         printf("%d ", arr[c]);
17     }
18 }
19 int* crop(int* arr){
20     arr = arr;
21     return arr;
22 }
23
24 int main()
25 {
26     // Question 7
27     // 20MAI1001 PRABHU
28     int newArr[100] = { 0 };
29     int arr1[] = {8,53,87,23,6,3,10,20,30,40};
30     int* arr = arr1;
31     int n = sizeof(arr1)/sizeof(arr1[0]);
32     int newPos = 0;
33     int dup_n = n;
34
35     for (int i = 0; i < n; i++) {
36         int head = *arr;
37         printf("%d'th iteration - ",i);
38         printf("dep_n = %d\n",dup_n);
39     }
40
41     while (dup_n > 0) {
42         //printf(arr,dup_n);
43     }
44
45     newArr[newPos] = head;
46     newPos++;
47     head = 0;
48     dup_n--;
49     //crop(arr);
50     printf("Input array - ");
51     printf(arr,dup_n);
52     arr++;
53     printf("\n");
54     printf("New array - ");
55     printf(newArr,n-dup_n);
56     printf("\n\n");
57 }
58
59 printf("Final non-decreasing array - ");
60 printf(newArr,n);
61 return 0;
62 }

```

Output:

```

$gcc -o main *.c
$main
0'th iteration - dep_n = 10
Input array - 3 53 87 23 6 10 20 30
New array - 3
1'th iteration - dep_n = 9
Input array - 6 87 53 23 8 10 20 30
New array - 3 6
2'th iteration - dep_n = 8
Input array - 8 87 53 23 10 20 30
New array - 3 6 8
3'th iteration - dep_n = 7
Input array - 10 87 53 23 20 30
New array - 3 6 8 10
4'th iteration - dep_n = 6
Input array - 20 87 53 23 30
New array - 3 6 8 10 20
5'th iteration - dep_n = 5
Input array - 23 87 53 30
New array - 3 6 8 10 20 23
6'th iteration - dep_n = 4
Input array - 30 87 53
New array - 3 6 8 10 20 23 30
7'th iteration - dep_n = 3
Input array - 40 87
New array - 3 6 8 10 20 23 30 40
8'th iteration - dep_n = 2
Input array - 53
New array - 3 6 8 10 20 23 30 40 53
9'th iteration - dep_n = 1
Input array -
New array - 3 6 8 10 20 23 30 40 53 87
Final non-decreasing array - 3 6 8 10 20 23 30 40 53 87

```


8)

// Online C compiler to run C online.

// Write C code in this online editor and run it.

#include <stdio.h>

```

void removeDup(int* a,int n) {
    for(int i = 0; i < n; i++)
    {
        for(int j = i+1; j < n; )
        {
            if(a[j] == a[i])
            {
                for(int k = j; k < n; k++)
                {
                    a[k] = a[k+1];
                }
                n--;
            }
            j++;
        }
    }

    for(int i = 0; i < n; i++)
    {
        printf("%d ", a[i]);
    }
}

```

```

int main()
{
    // Question 8
    // 20MAI1001 PRABHU
    int newArr[100] = { 0 };
    int arr1[]={8,53,87,53,23,6,3,8,10,20,30,40,53};
    int* arr = arr1;
    int n = sizeof(arr1)/sizeof(arr1[0]);
    int newPos = 0;
    int dup_n = n;
    int posHead = 0;
    int head;

    //flag = true;
    for (int i = 0; i < n; i++) {
        head = *(arr+i);
    }
}

```

```

        posHead = i;
        for(int j = i+1; j<n; j++) {
            if(*(arr+j) == head) {
                if(j > 2*posHead) {
                    newArr[newPos] = *(arr+j);
                    newPos++;
                    break;
                }
            }
            else{
                head = *(arr+j);
                posHead = j;
            }
        }
    }
}

printf("Weaklings in the array - ");
removeDup(newArr,newPos);
//printf("%d",newArr[newPos]);

return 0;
}

```

The screenshot shows an online C compiler interface with the following components:

- Browser Tabs:** Database Courses, Introduction to Te..., *Face Mask Detect..., Untitled5.ipynb, Database System, DS4_A01.qxd, notepad.pw / pgs..., Online C Compiler.
- Address Bar:** tutorialspoint.com/compile_c_online.php
- Page Header:** codingground, Compile and Execute C Online (GNU GCC v7.1.1), Fork, Project, Edit, Setting, Login.
- Code Editor:**

```

1 // Online C compiler to run C online.
2 // Write C code in this online editor and run it.
3
4 #include <stdio.h>
5
6 void removeDup(int* a,int n) {
7     for(int i = 0; i < n; i++)
8     {
9         for(int j = i+1; j < n; )
10        {
11            if(a[j] == a[i])
12            {
13                for(int k = j; k < n; k++)
14                {
15                    a[k] = a[k+1];
16                }
17                n--;
18                j++;
19            }
20        }
21    }
22
23    for(int i = 0; i < n; i++)
24    {
25        printf("%d ", a[i]);
26    }
27 }
28
29 int main()
30 {
31     // Question 8
32     // 20MAI1001 PRABHU
33     int newArr[100] = { 0 };
34     int arr1[]={8,53,87,53,23,6,3,8,10,20,30,40,53};
35     int* arr = arr1;
36     int n = sizeof(arr1)/sizeof(arr1[0]);
37     int newPos = 0;
38     int dup_n = n;

```
- Result Panel:**

```

$gcc -o main *.c
$main
Weaklings in the array - 8 53

```

