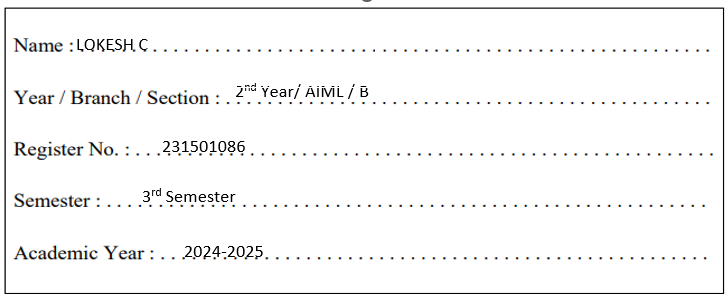
**RAJALAKSHMIENGINEERINGCOLLEGE**

**RAJALAKSHMI NAGAR, THANDALAM – 602 105**



|  |
| --- |
| **CS23331 - DESIGN AND ANALYSIS OF ALGORITHM** |
| **LABORATORY LAB MANUAL** |



**WEEK**

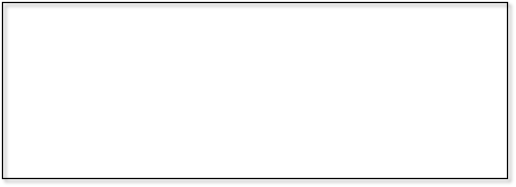
**–**

**05**

**PLAYING**

**WITH**

**NUMBERS**



|  |  |
| --- | --- |
| **EXPERIMENT NO : 5.1** | **DATE :** |
| **REGISTER NO : 231501018** | **NAME : ANYA M** |



# PLAYING WITH NUMBERS

**PLAYING WITH NUMBERS:**

RAM AND SITA ARE PLAYING WITH NUMBERS BY GIVING PUZZLES TO EACH

OTHER. NOW IT WAS RAM TERM, SO HE GAVE SITA A POSITIVE INTEGER ‘N’ AND

TWO NUMBERS 1 AND 3. HE ASKED HER TO FIND THE POSSIBLE WAYS BY WHICH THE NUMBER N CAN BE REPRESENTED USING 1 AND 3.WRITE ANY EFFICIENT ALGORITHM TO FIND THE POSSIBLE WAYS.

**EXAMPLE 1:**

**INPUT:**

6

**OUTPUT:**

6

**EXPLANATION:**

THERE ARE 6 WAYS TO 6 REPRESENT NUMBER WITH 1 AND 3

1+1+1+1+1+1

3+3

1+1+1+3

1+1+3+1

1+3+1+1

3+1+1+1

**INPUT FORMAT**

FIRST LINE CONTAINS THE NUMBER N

**OUTPUT FORMAT**

**PRINT:**

THE NUMBER OF POSSIBLE WAYS ‘N’ CAN BE REPRESENTED USING 1 AND 3

**SAMPLE INPUT**

6

**SAMPLE OUTPUT**

6

**PROGRAM**

#include

<

stdio.h

>

int

main

()

{

long

n;

scanf

(

"

%ld

"

,

&

n);

if

(

n

<

0

)

{

return

0

;

}

long

array

[

n

+

1

]

;

array

[

0

]

=

1

;

array

[

1

]

=

1

;

array

[

2

]

=

1

;

array

[

3

]

=

2

;

for

(

long

i

=

4

;

i

<=

n;

i

++

)

{

array

]

[

i

=

array

[

i

-

1

]

+

array

[

i

-

3

]

;

}

printf

(

"

%ld

\n

"

,

array

[

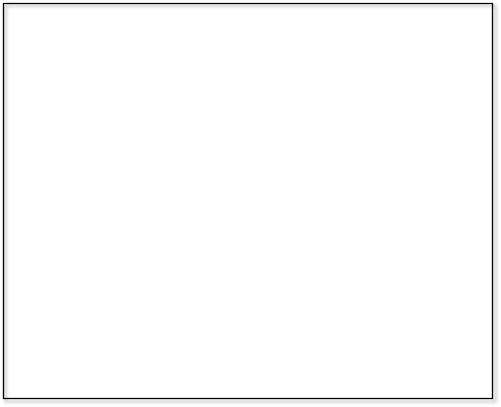
n]);

return

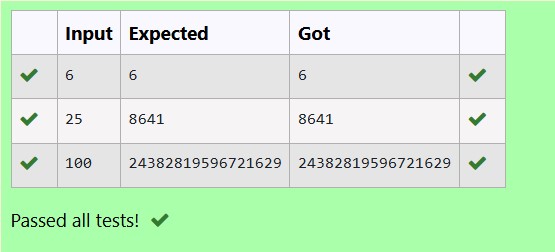
0

;

}



**OUTPUT**



|  |  |
| --- | --- |
| **EXPERIMENT NO : 5.2** | **DATE :** |
| **REGISTER NO : 231501018** | **NAME : ANYA M** |



# PLAYING WITH CHESSBOARD

**PLAYING WITH CHESSBOARD:**

RAM IS GIVEN WITH AN N\*N CHESSBOARD WITH EACH CELL WITH A MONETARY

VALUE. RAM STANDS AT THE (0,0), THAT THE POSITION OF THE TOP LEFT WHITE

ROOK. HE IS BEEN GIVEN A TASK TO REACH THE BOTTOM RIGHT BLACK ROOK

POSITION (N-1, N-1) CONSTRAINED THAT HE NEEDS TO REACH THE POSITION BY

TRAVELING THE MAXIMUM MONETARY PATH UNDER THE CONDITION THAT HE CAN ONLY TRAVEL ONE STEP RIGHT OR ONE STEP DOWN THE BOARD. HELP RAM TO ACHIEVE IT BY PROVIDING AN EFFICIENT DP ALGORITHM.

**EXAMPLE:**

**INPUT**

3

1. 2 4
2. 3 4

8 7 1

**OUTPUT:**

19

**EXPLANATION:**

TOTALLY THERE WILL BE 6 PATHS AMONG THAT THE OPTIMAL IS

OPTIMAL PATH VALUE:1+2+8+7+1=19

**INPUT FORMAT**

* FIRST LINE CONTAINS THE INTEGER N
* THE NEXT N LINES CONTAIN THE N\*N CHESSBOARD VALUES

**OUTPUT FORMAT**

PRINT MAXIMUM MONETARY VALUE OF THE PATH

**PROGRAM**



**OUTPUT**



|  |  |
| --- | --- |
| **EXPERIMENT NO : 5.3** | **DATE :** |
| **REGISTER NO : 231501018** | **NAME : ANYA M** |



**LONGEST COMMON SUBSEQUENCE**

GIVEN TWO STRINGS FIND THE LENGTH OF THE COMMON LONGEST SUBSEQUENCE(NEED NOT BE CONTIGUOUS) BETWEEN THE TWO.

**EXAMPLE:**

**S1:** GGTABE

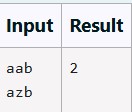
**S2:** TGATASB

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S1: | A | G | G | T | A | B |  |
| S2: | G | X | T | X | A | Y | B |

THE LENGTH IS 4

**SOLVING IT USING DYNAMIC PROGRAMMING**

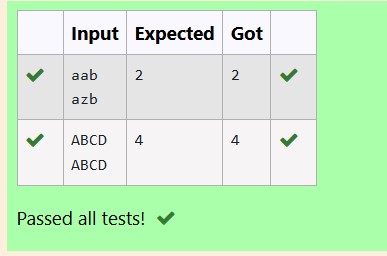
**FOR EXAMPLE:**



**PROGRAM**



**OUTPUT**



|  |  |
| --- | --- |
| **EXPERIMENT NO : 5.4** | **DATE :** |
| **REGISTER NO : 231501018** | **NAME : ANYA M** |



**LONGEST NON-DECREASING SUBSEQUENCE**

**PROBLEM STATEMENT:**

FIND THE LENGTH OF THE LONGEST NON-DECREASING SUBSEQUENCE IN A GIVEN SEQUENCE.

**EXAMPLE:**

**INPUT:**

#include

>

<

stdio.h

int

longseq

(

int

arr

[]

,

int

n

){

int

dp

[

n];

for

(

int

i

=

0

;i

<

n;i

++

){

dp

]

[

i

=

1

;

}

for

(

int

i

=

1

;

i

<

n;

i

++

)

{

for

(

int

j

=

0

;

j

<

i;

j

++

)

{

if

(

arr

[

i

]

>=

arr

[

j

])

{

dp

i

]

[

=

(

dp

[

i

]

>

dp

[

j

]

+

1

)

?

dp

i

]

[

:

dp

[

j

]

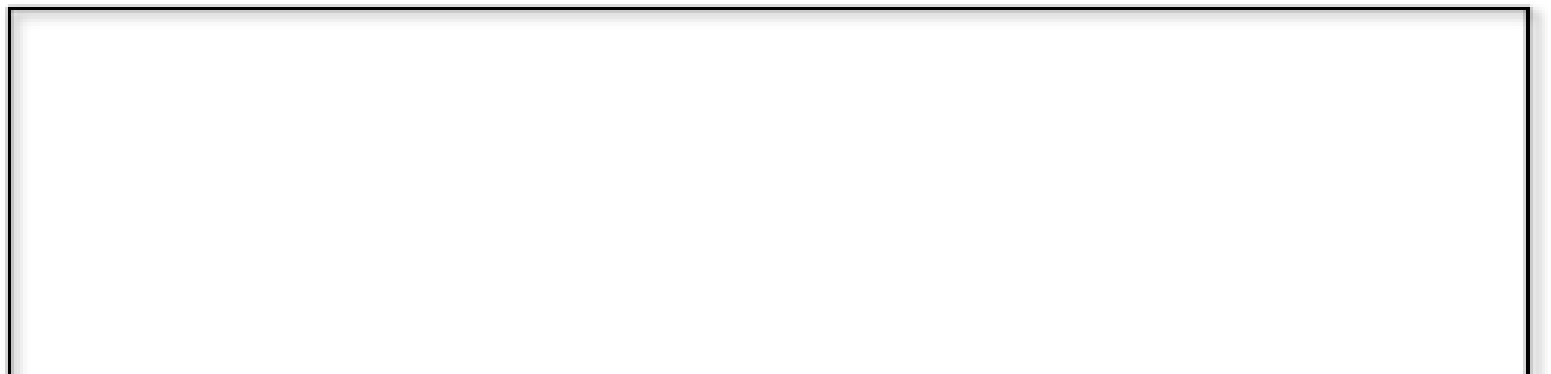
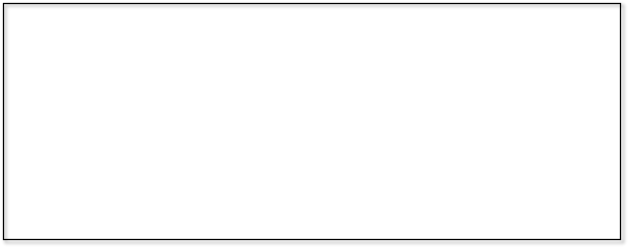
+

1

;

}

}



9

**SEQUENCE:[-1,3,4,5,2,2,2,2,3] THE SUBSEQUENCE IS [-1,2,2,2,2,3]**

**OUTPUT:**

6

# PROGRAM

} int maximumlength=0; for(int i=0;i<n;i++){ if(dp[i]>maximumlength){ maximumlength=dp[i];

}

}

return maximumlength;

}

int main()

{

int n;

scanf("%d",&n);

int arr[n]; for(int i=0;i<n;i++)

{

scanf("%d",&arr[i]);

}

int length=longseq(arr, n);

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

return 0;

}

**OUTPUT**

