Name : Mukul Kumar

Enrollment number : 201B162

Batch : B5

Subject : OOPL

Lab number : 2

Semester : 2

1. Write a function that will return the length of a character string using pointer. You are

not allowed to use the strlen C library function.

Code:

#include <stdio.h>

int stringlength(char \*a)

{

int i;

char check = \*a;

for (i = 0; check != '\0'; i++)

{

check = \*(a + i + 1);

}

return i;

}

int main()

{

char str[20] ;

scanf("%s",str);

printf("%d", stringlength(str));



2. Write a function that finds the minimum and the maximum value in an array of N

integers. Inputs to the function are the array of integers, an integer variable containing

the length of the array and pointers to integer variables that will contain the minimum

and the maximum values.

The function prototype is:

void minmax( int array[], int length, int \* min, int \* max);

Write a main function that uses this function to find and display the minimum and the

maximum values of an array of integers.

Code:

#include <stdio.h>

Void minmax(int array[], int length, int \*min, int \*max);

Int main()

{

Int N;

Scanf(“%d”, &N);

Int arr[N], min = 0, max = 0;

For (int I = 0; I < N; i++)

Scanf(“%d”, &arr[i]);

Minmax(arr, N, &min, &max);

Printf(“min : %d\nmax : %d”, min, max);

Return 0;

}

Void minmax(int array[], int length, int \*min, int \*max)

{

\*max = array[0];

\*min = array[0];

For (int I = 0; I < length; i++)

{

If (array[i] > \*max)

\*max = array[i];

}

For (int I = 0; I < length; i++)

{

If (array[i] < \*min)

\*min = array[i];

}

}



3. Write a menu driven program (using switch-case) to create a database of student names and

perform the following operations using array of character pointers and dynamic memory allocation.

(A) To insert a student name

(B) To delete a name (Show Error message if zero names are there in database)

(C) To print the names

Note: your program should keep on showing above three options until user enters ‘‘N’.

Code:

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <stdlib.h>

char \*string\_input(int \*);

typedef struct arr

{

char \*pointer\_to\_char;

int size;

} arr;

int main()

{

int count = 0;

char x = 'A';

arr \*A;

A = (arr \*)malloc(1 \* sizeof(arr));

while (x)

{

printf("A : Enter name\nB : Delete name\nC : Print all names\nE : Clear screen\nN : Exit program\n\n");

x = getch();

printf("%c\n", x);

switch (x)

{

case 'A':

{

printf("Enter your name : \n");

int size;

(A + count)->pointer\_to\_char = string\_input(&size);

(A + count)->size = size - 1;

count++;

A = (arr \*)realloc(A, (count + 1) \* sizeof(arr));

printf("\"%s\" registered successfully\n\n", (A + count - 1)->pointer\_to\_char);

break;

}

case 'B':

{

printf("Enter name you want to delete\n");

int del = 0, size;

char \*str = (string\_input(&size));

if (count == 0)

printf("memory Error-database empty\n\n");

else

{

while (del != count)

{

if (del == count)

printf("not found - please enter a valid data");

if (strcmp(str, (A + del)->pointer\_to\_char) == 0)

{

while (del != count - 1)

{

(A + del)->pointer\_to\_char = (A + del + 1)->pointer\_to\_char;

del++;

}

A = (arr \*)realloc(A, (count - 1) \* sizeof(arr));

count--;

printf("data deleted successfully\n");

break;

}

del++;

}

}

break;

}

case 'C':

{

printf("You have entered :\n");

if (count != 0)

for (int i = 0; i < count; i++)

{

printf("%s\n", (A + i)->pointer\_to\_char);

if (i == count - 1)

{

printf("\n");

}

}

else

printf("Nothing\n\n");

break;

}

case 'E':

{

clrscr();

}

case 'N':

{

return 0;

}

default:

printf("inputError-Invalid input\n\n");

}

}

}

char \*string\_input(int \*size)

{

char \*p, ch;

int len = 1;

p = (char \*)malloc(len \* sizeof(char));

while ((scanf("%c", &ch)) && ch != '\n')

{

\*(p + len - 1) = ch;

len++;

p = (char \*)realloc(p, len \* sizeof(char));

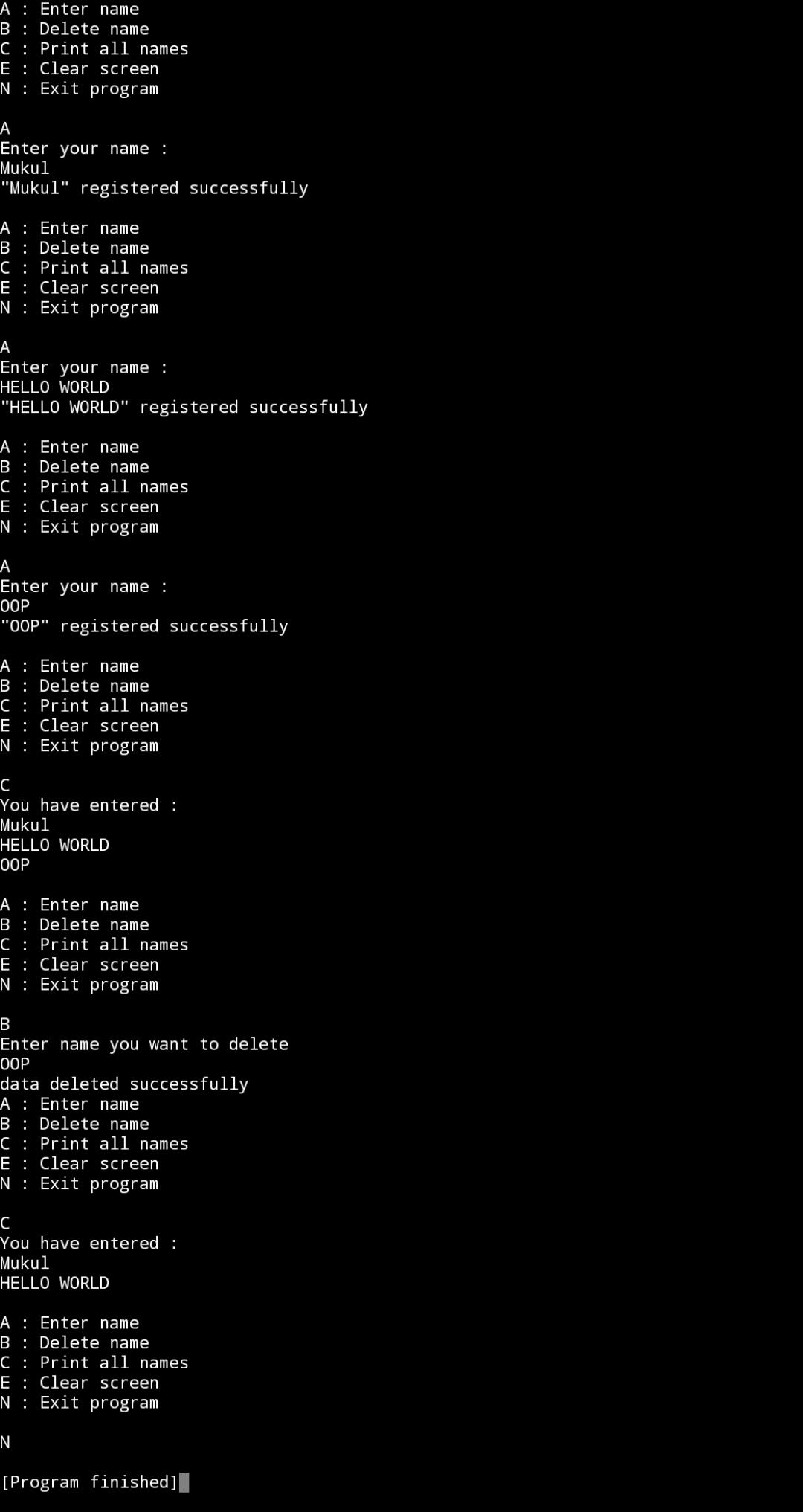
}

\*size = len;

\*(p + len - 1) = '\0';

return p;

}



4. Write a program to generate random numbers in given range [m, n].

Test case :

Input: m=10, n=50

Output: 34

Code :

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

int main()

{

int m,n;

printf("Input lower range and upper range :: ");

scanf("%d %d",&m,&n);

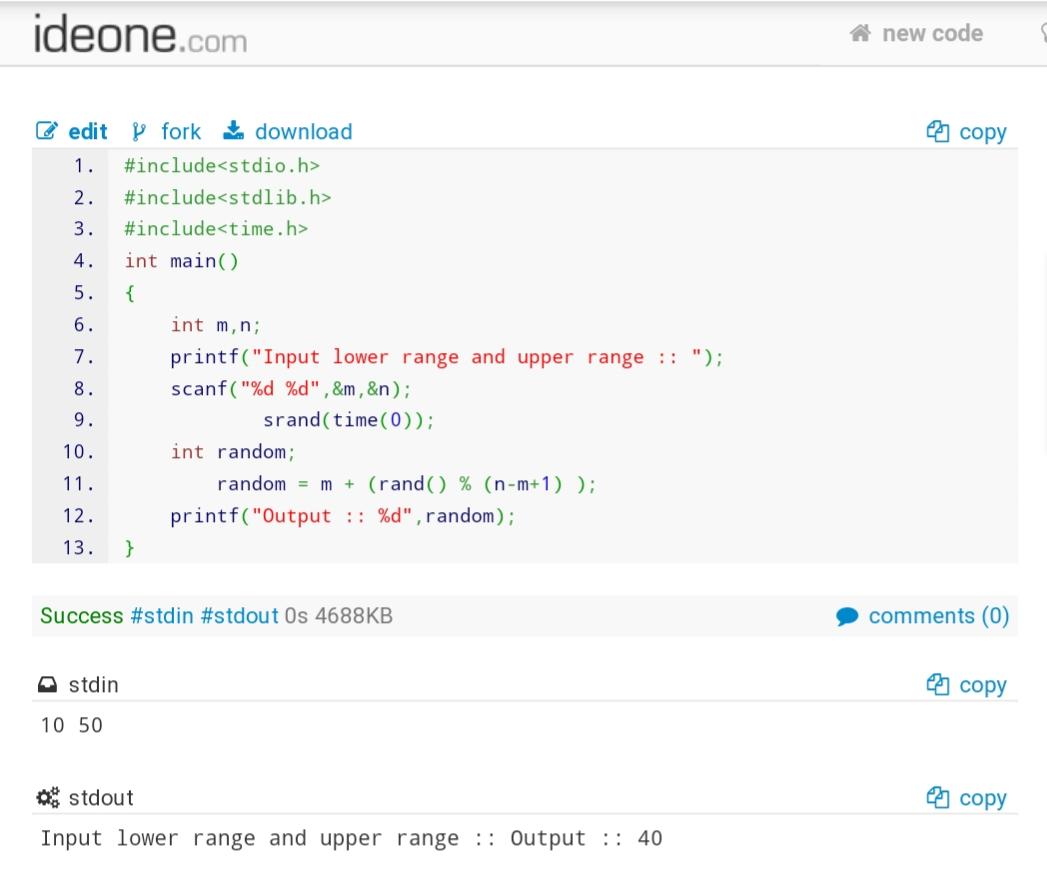
srand(time(0));

int random;

random = m + (rand() % (n-m+1) );

printf("Output :: %d",random);

}



5. An automobile company has serial number for engine parts starting from AA0 to FF9.

The other characteristics of parts to be specified in a structure are: Year of manufacturing, material (Steel, Aluminum, Iron etc.) and quantity manufactured.

Write a program to include following tasks:

(a) Specify a C structure that includes four members viz. serial numbers, year of manufacturing, material type, and quantity. Declare an array of structure of size

60.

(b) Automatically initialize the values of structure members as follows :

i. Generate serial numbers such that first part has serial number AA0 and 60th part has serial number FF9.

ii. Randomly generate year of manufacturing in the range [1990-2018]

iii. Randomly initialize material type from three choices ie. Steel, Aluminum, Iron

iv. Randomly initialize quantity in the range [1-1000]

(c) Display the information of the parts with serial numbers between any given range such as [BB1,CC6].

Code :

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <time.h>

Struct automobile

{

Char serialnumber[4];

Int year\_of\_manafacturing;

Char materialtype[10];

Int quantity;

};

Int main()

{

Srand(time(0));

Struct automobile car1[60];

Char serial\_identity[6] = {‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’};

Char material\_type[3][10] = {“Steel”, “Aluminium”, “Iron”};

For (int I = 0; I < 6; i++)

{

For (int j = 0; j < 10; j++)

{

Car1[(10 \* i) + j].serialnumber[0] = serial\_identity[i];

Car1[(10 \* i) + j].serialnumber[1] = serial\_identity[i];

Car1[(10 \* i) + j].serialnumber[2] = j+48;

Car1[(10 \* i) + j].serialnumber[3] = ‘\0’;

Car1[(10 \* i) + j].year\_of\_manafacturing = 1990 + (rand() % (2018 – 1990 + 1));

Strcpy(car1[(10 \* i) + j].materialtype, material\_type[0 + (rand() % (2 + 1))]);

Car1[(10 \* i) + j].quantity = 1 + (rand() % (1000 – 1 + 1));

}

}

For (int I = 1; I <= 2; i++)

For (int j = 0; j < 10; j++)

{

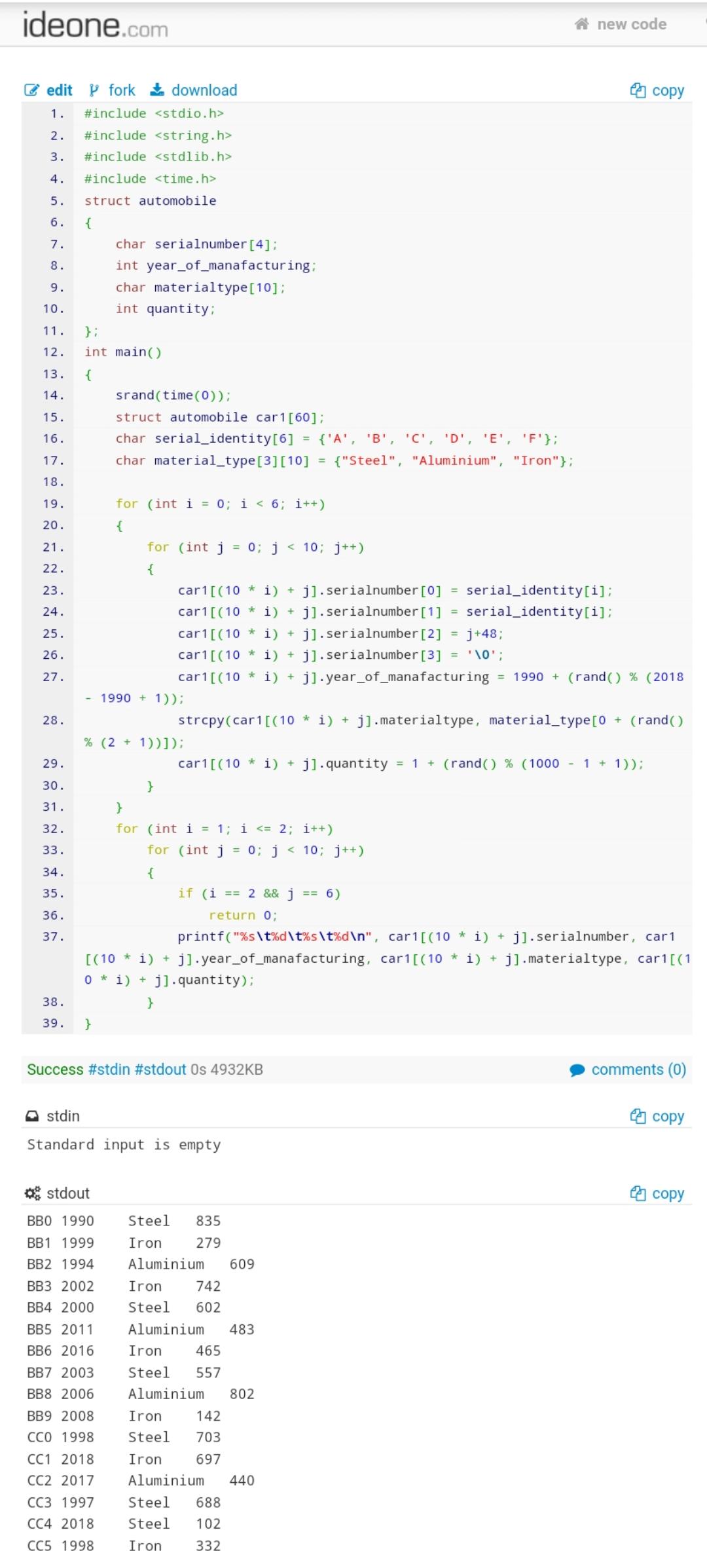
If (I == 2 && j == 6)

Return 0;

Printf(“%s\t%d\t%s\t%d\n”, car1[(10 \* i) + j].serialnumber, car1[(10 \* i) + j].year\_of\_manafacturing, car1[(10 \* i) + j].materialtype, car1[(10 \* i) + j].quantity);

}

}



1. Given an array A of size N-1 and given that there are numbers from 1 to N with one element missing; Write program to find the missing number.

Test case 1: Given array: 1 2 3 5; missing element is 4.

Test case 2: Given array: 1 2 3 4 5 6 7 8 10; missing element is 9.

Code :

#include <stdio.h>

int getMissingNo(int a[], int n)

{

int sum = 0 ;

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

{

sum += a[i];

}

return ((n\*(n+1))/2)-sum;

}

int main()

{

int n;

printf("ENTER THE SIZE OF ARRAY \n");

scanf("%d",&n);

int arr[n] ;

printf("ENTER THE VALUES ");

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

{

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

}

printf("%d",getMissingNo( arr, n));

}

