CONTENTS

EX.NO: 01 **/*SUM OF INDIVIDUAL DIGIT*/**

DATE:

AIM:

To Test the C program: Finding the sum of individual digits of a 10-digit number until a single digit is produced.

ALGORITHM:

Step 1:Start the process
Step 2:Get number by user.
Step 3:Get the modulus/remainder of the number.
Step 4:sum the remainder of the number.
Step 5:Divide the number by 10.
Step 6:Repeat the step 2 while number is greater than 0.
Step 7:Display the sum of digits
Step 8:Stop the process

PROGRAM 1:

```
#include<stdio.h>
int main()
{
Long num;
int dig,sum;
printf("Enter the number : ");
scanf("%ld",&num);
printf("%ld-> ",num);
do
{
sum = 0;
while(num!=0)
{
dig=num%10;
sum+=dig;
num/=10;
}
printf("%d-> ",sum);
num=sum;
}
while(num/10!=0);
return 0;
}
```

TEST CASE:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC01	CHECKING THEPROCESS FOR INPUT & OUTPUT	ADD THE HEADER FILE STATEMENT #INCLUDE<STDIO .H>	DISPLAY INPUT& OUTPUT STATEMENT PROPERLY	DISPLAYE D INPUT & OUTPUT STATEMEN TS PROPERLY	SUCCESS
TC02	SUCCESSFU LPROCESS OF MAIN FUNCTION	MAIN FUNCTION STATEMENT SHOULD BE CHECKED	MAIN FUNCTION IS ALLOWING THE PROGRAM TO PROCESS	MAIN FUNCTION ALLOWED THE PROGRAM TO PROCESS	SUCCESS
TC03	INITIATION OFVARIABLE NAME NUM	INITIATE THE VARIAVBLE NUMWITH THE DATA TYPE LONG	ACCEPTAN CEOF LONG INTEGER VARIABLE VALUES	NOT ACCEPTAN CEOF LONG INTEGER VARIABLE VALUES	FAILURE
TC04	GETTING INPUTVALUES FROM THE USER	ENTER INPUT TO CALCULATE SUM OFDIGIT	ACCEPT THE INPUT FROMTHE USER	ACCEPT EDTHE INPUT FROM THE USER	SUCCESS
TC05	CHECK THE GIVEN INPUT NUMBER WHETHER IT ISCORRECT OR NOT	ENTER THE INPUT VALUE AS INTEGER	ACCEPT AND DISPLAY THE GIVEN INPUT VALUE	NOT ACCEPTED AND DISPLAY THE GIVEN INPUT VALUE	FAILURE
TC06	INITIATE THE PROCESS OF DOLOOP	PROCESSING SUMOF DIGIT	DISPLAY THE SUM OF DIGIT	DISPLAYE D THE SUM OF DIGIT	SUCCESS
TC07	INITIATE THE VARIABLE VALUE OF SUM	GIVE THE INITIAL VALUE AS 0 TO SUM	DISPLAY THESUM VALUE AFTER SUMMING	DISPLAYE D THE SUM VALUE AFTER SUMMING	SUCCESS
TC08	INITIATE THEWHILE LOOPFOR PROCESSIN G	CHECK THE CONDITION FOR NUM VALUE IS NOTEQUAL TO ZERO	PROCESS THE SUM OF DIGITTO PRODUCE SINGLE DIGIT	PROCESSE D THE SUM OF DIGIT TO PRODUCE SINGLE DIGIT	SUCCESS
TC09	INITIATE VARIABLE VALUE OF DIG	DIVIDE THE NUM VALUE BY 10	ASSIGNED REMINDER VALUE IN DIG	ASSIGNED REMINDER VALUE IN DIG	SUCCESS

TC10	INITIATE THE VARIABLE VALUE OF NUM	DIVIDE THE NUM VALUE TO ASSIGN QUOTIENT VALUE	ASSIGNED THE QUOTIENT IN NUM VARIABLE	ASSIGNED THE QUOTIENT IN NUM VARIABLE	SUCCESS

ERROR CORRECTION:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC03	INITIATION OF VARIABLE NAME NUM	CHECK THE DATATYPE WITH THE LONG INTEGER	ACCEPTANCEOF LONG INTEGER VARIABLE VALUES	ACCEPTED THE VALUE OF LONG INTEGER	SUCCESS
TC05	CHECK THE GIVEN INPUT NUMBER WHETHER IT ISCORRECT OR NOT	GIVE THE CORRECT10 DIGIT INPUT VALUE AS INTEGER	ACCEPT AND DISPLAY THE GIVEN INPUT VALUE	ACCEPTED AND DISPLAYED THE GIVEN INPUT VALUE	SUCCESS

OUTPUT:

RESULT:

EX.NO: 02 **/*MARK LIST */**

DATE:

AIM:

To Test the C Program: Accept the inputs student name, marks in five subjects and declare the result as PASS if the student gets minimum 40 in each subject; otherwise declare the result as FAIL.

ALGORITHM:

Step 1 : Start the program
Step 2 : Assign 4 integer variables inputs for the different 4 subjects.
Step 3 : Then calculate the grade based upon the average of four marks .
Step 4 : If the value of the grade is more than 40, it will print pass otherwise it shows fail.
Step 5 : Run the program
Step 6 : stop the process

PROGRAM 2:

```
#include <stdio.h>
int main()
{
int ST,drawing, chemistry, math, c,total;
clrscr();
printf("\n Marks in:\n\n");
printf(" ST    = ");
scanf("%d",&ST);
printf(" drawing   = ");
scanf("%d",&drawing);
printf(" chemistry = ");
scanf("%d",&chemistry);
printf(" math      = ");
scanf("%d",&math);
printf(" c         = ");
scanf("%d",&c);
total=math+ST+chemistry+drawing+c;
if(ST<40 || chemistry<40 || drawing<40 || math<40 || c<40)
printf("\n Result: FAIL");
else
printf("\n Result: PASS");
printf("\n  Percentage:%0.2f\n",total/5.0);
printf("\n NOTE: Total for each subject is 100\n");
return 0;
}
```

TEST CASE:

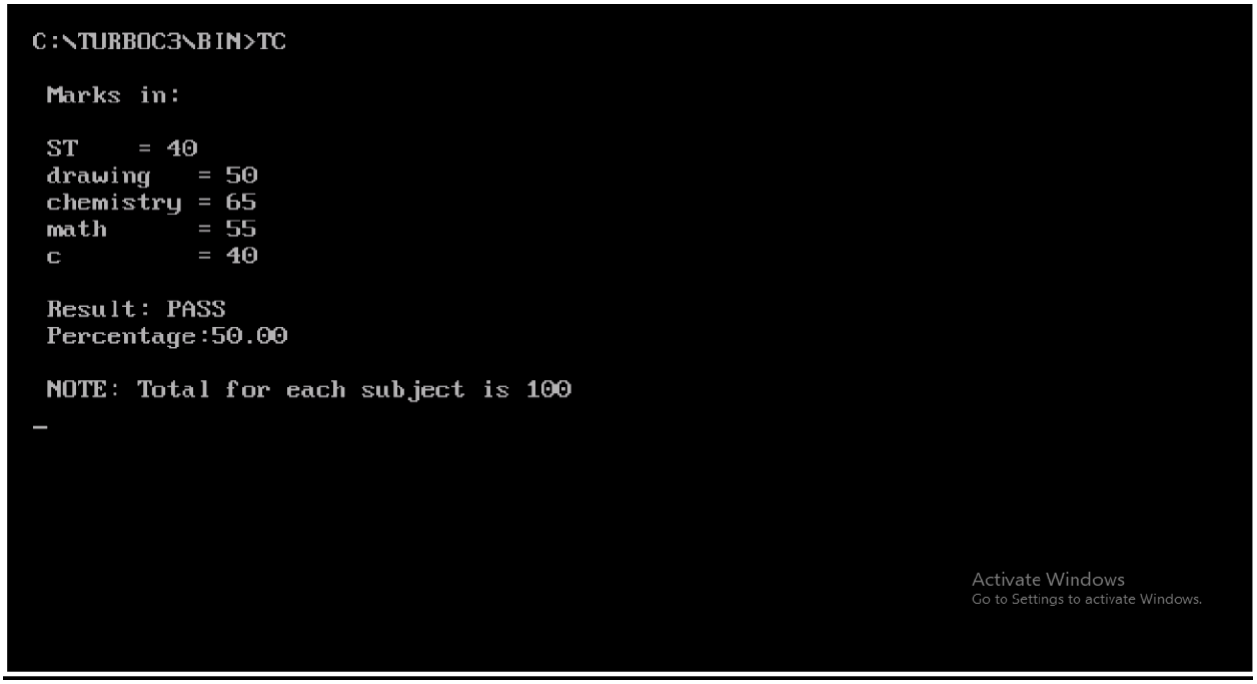
TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC01	CHECKING THEPROCESS FOR INPUT & OUTPUT	ADD THE HEADER FILE STATEMENT #INCLUDE<STDIO.H>	DISPLAY INPUT& OUTPUT STATEMENTS PROPERLY	DISPLAYED INPUT & OUTPUT STATEMENTS PROPERLY	SUCCESS
TC02	SUCCESSFULPROCESS OFMAIN FUNCTION	MAIN FUNCTION STATEMENT SHOULD BE CHECKED	MAIN FUNCTION IS ALLOWING THE PROGRAM TO PROCESS	MAIN FUNCTION ALLOWED THE PROGRAM TO PROCESS	SUCCESS
TC03	INITIATION OF VARIABLES TOGET INPUT VALUES	INITIATE THE VARIABLES WITHTHE DATA TYPE INTEGER	ACCEPTANCEOF INTEGER VARIABLE VALUES	ACCEPTED INTEGER VARIABLE VALUES	SUCCESS
TC04	GETTING INPUTVALUES FROM THE USER	ENTER INPUT TO CALCULATE TOTAL MARKS OF GIVEN INPUT	ACCEPT THE INPUT FROM THE USER	ACCEPTEDTHE INPUT FROM THE USER	SUCCESS
TC05	ASSIGN THE TOTAL MARKSIN THE VARIABLE TOTAL	ADD THE FIVE SUBJECTS MARKS	ASSIGN THE ADDED VALUESIN THE VARIABLE TOTAL	ASSIGNED THEADDED VALUES IN THE VARIABLE TOTAL	SUCCESS
TC06	CHECK THE GIVEN INPUT MARKS ABOVE40	CHECK THE CONDITION USINGOR OPERATOR	ALL THE GIVEN VALUES HAS TO BE CHECKED USING OR OPERATOR	ALL THE GIVEN VALUES SUCCESSFULLYCHECKED USING OR OPERATOR	SUCCESS
TC07	PRINT THE RESULT AS PASS	CHECK ALL THE GIVEN MARKS ARE ABOVE 40	USING OR OPERATOR GIVEN INPUT WILL BE CHECKED	IDENTIFIED ONE OF THE GIVEN INPUTIS BELOW 40	FAILURE
TC08	PRINT THE RESULT AS FAIL	CHECK GIVEN MARKS ARE BELOW40	USING OR OPERATOR GIVEN INPUT WILL BE CHECKED	IDENTIFIED ONE OF THE GIVEN INPUTIS BELOW 40	SUCCESS

TC09	PRINT THE PERCENTAGE	CALCULATE THE TOTAL MARKS FOR ALL SUBJECTS	ASSIGN THE CALCULATED TOTAL MARKS AND DIVIDE MARKS USING NUMBER SYSTEM	PRINTED THE PERCENTAGEVALUE	SUCCESS
TC10	GIVE THE IDENTIFICATIONFOR MAXIMUM MARKS FOR EACH SUBJECT	USE PRINTF STATEMENT FOR IDENTIFICATION	PRINT THE MAXIMUM MARK FOR EACH SUBJECT IS 100	PRINTED THE MAXIMUM MARK FOR EACH SUBJECT IS 50	FAILURE

ERROR CORRECTION:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC07	PRINT THE RESULT AS PASS	CORRECT INPUT HAS TO BE GIVEN	USING OR OPERATOR GIVEN INPUT WILL BE CHECKED	PRINTED THE RESULT AS PASS	SUCCESS
TC10	GIVE THE IDENTIFICATIONFOR MAXIMUM MARKS FOR EACH SUBJECT	USE PRINTF STATEMENT FOR IDENTIFICATION AND CHECK THE TYPING MISTAKES IN KEYBOARD	PRINT THE MAXIMUM MARK FOR EACH SUBJECTIS 100	PRINTED THE MAXIMUM MARK FOR EACH SUBJECT IS 50	SUCCESSS

OUTPUT:



RESULT:

EX.NO: 03 **/*PRIME NUMBERS */**

DATE:

AIM:

To Test the C program: Program for generating n prime numbers

ALGORITHM:

Step 1: Start the program
Step 2: First find the factors of the given number.
Step 3: Check the number of factors of that number.
Step 4: If the number of factors is more than two, it is not a prime number.
Step 5: Run the program
Step 6: Stop the program

PROGRAM 3:

```
#include <stdio.h>
int main()
{
int i, j, end, isPrime;
printf("Find prime numbers between 1 to : ");
scanf("%d", &end);
printf("All prime numbers between 1 to %d are:\n", end);
for(i=2; i<=end; i++)
{
isPrime = 1;
for(j=2; j<=i/2; j++)
 {
if(i%j==0)
 {
isPrime = 0;
break;
}
}
if(isPrime==1)
{
printf("%d, ", i);
}
}
return 0;
}
```

TEST CASE:

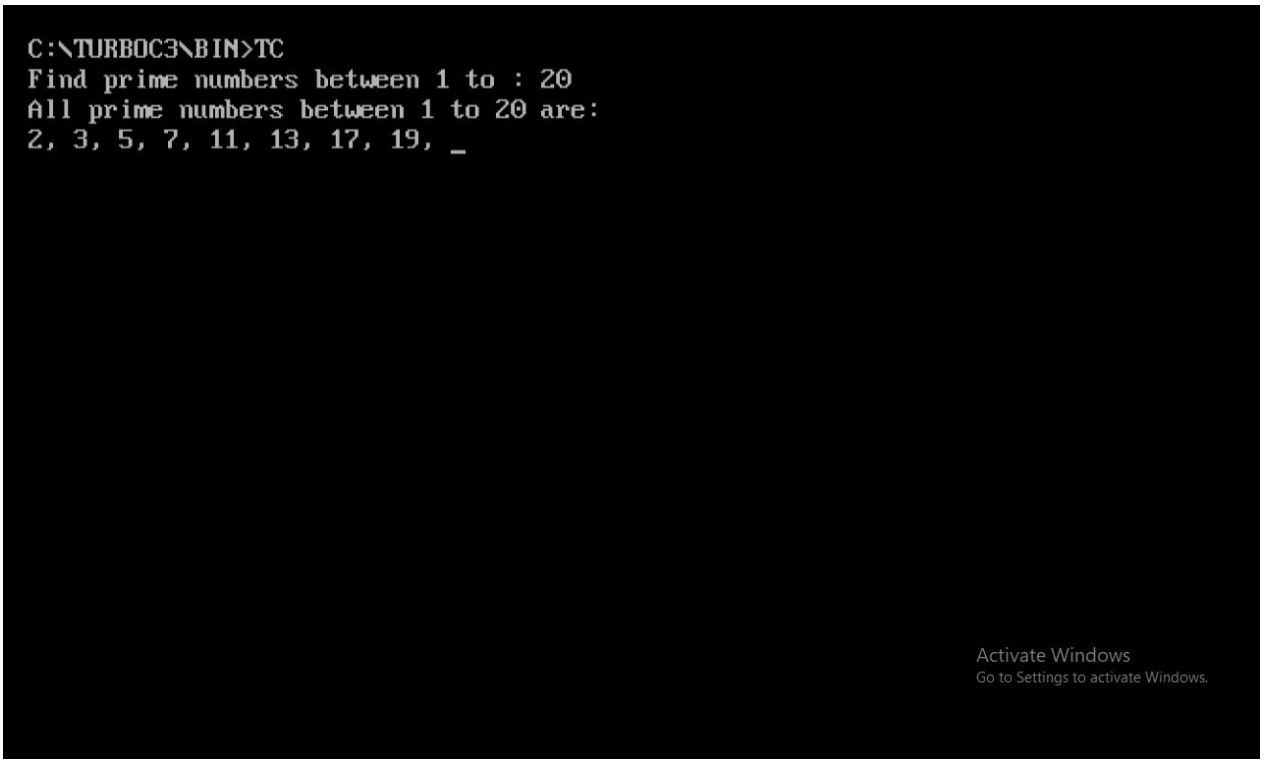
TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC01	CHECKING THEPROCESS FOR INPUT & OUTPUT	ADD THE HEADER FILE STATEMENT #INCLUDE<STDIO .H>	DISPLAY INPUT& OUTPUT STATEMENT S PROPERLY	DISPLAYE D INPUT & OUTPUT STATEMEN TS PROPERLY	SUCCESS
TC02	SUCCESSFU LPROCESS OFMAIN FUNCTION	MAIN FUNCTION STATEMENT SHOULD BE CHECKED	MAIN FUNCTION IS ALLOWING THE PROGRAM TO PROCESS	MAIN FUNCTION ALLOWED THE PROGRAM TO PROCESS	SUCCESS
TC03	INITIATION OF VARIABLES TOGET INPUT VALUES	INITIATE THE VARIABLES WITHTHE DATA TYPE INTEGER	ACCEPTAN CEOF INTEGER VARIABLE VALUES	NOT ACCEPT ED INTEGER VARIAB LE VALUES	FAILURE
TC04	GET INPUT VALUES FROMTHE USER	ENTER INPUT RANGE TO FINDPRIME NUMBER	ACCEPT THE INPUT RANGE FROM THE USER	ACCEPTED THE INPUT RANGE FROMTHE USER	SUCCESS
TC05	ASSIGN THE INPUT VALUETO THE VARIABLE ISPRIME	ASSIGN THE INPUT VALUE AS 1 FOR THE VARIABLE ISPRIME	DISPLAY THEFIRST PRIME NUMBER ASONE	DISPLAYE D THE FIRST PRIME NUMBER AS TWO	FAILURE
TC06	CHECKING THEPROCESS FOR PRIME OR NOT	USE FOR LOOP TOFIND PRIME NUMBERS	DISPLAY THE PRIME NUMBERS IF ITIS FOUND	DISPLAY EDTHE PRIME NUMBER S	SUCCESS
TC07	CHECKING THEPROCESS FOR PRIME OR NOT	CHECK THE FOR LOOP VARIABLE I ISDIVISIBLE BY ANY NUMBER OTHER THAN 1 AND SELF	DISPLAY THE PRIME NUMBERS IF IT IS NOT DIVISIBLE BY ANY NUMBER OTHER THAN 1AND SELF	DISPLAY EDTHE PRIME NUMBER S	SUCCESS
TC08	BREAK THE PROCESS IF IT IS NOT PRIME	CHECK THE DIVISIBLE VALUE I%J =0 OR NOT	IF IT IS ZERO THE PROCESS WILL BE QUIT	QUIT THE PROCES S	SUCCESS
TC09	PRINT THE PRIME NUMBER	PRINT THE VARIABLE OF I VALUE AS PRIME	DISPLAY THE PRIME NUMBERS	DISPLAY THE PRIME NUMBER	SUCCESS

		NUMBER			
TC10	BREAK THE PROCESS	FIND THE PRIME NUMBERS REACHEDTHE LIMIT VALUE	QUIT THE PROCESS IF ITIS REACH	QUIT THE PROCES	SUCCESS

ERROR CORRECTION:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC03	INITIATION OF VARIABLES TOGET INPUT VALUES	CHECK THE INITIATED VARIABLES WITHTHE DATA TYPE INTEGER	ACCEPTAN CEOF INTEGER VARIABLE VALUES	ACCEPTED INTEGER VARIABLE VALUES	SUCCESS
TC05	ASSIGN THE INPUT VALUETO THE VARIABLE ISPRIME	CHECK THE ASSIGNED INPUTVALUE AS 1 FOR THE VARIABLE ISPRIME	DISPLAY THEFIRST PRIME NUMBER ASONE	DISPLAYED THE FIRST PRIME NUMBER AS ONE	SUCCESS

OUTPUT :



RESULT:

EX.NO: 04 **/*SORTING ARRAYS*/**

DATE:

AIM:

To Test the C program: Sort and store the elements of two arrays of integers into the third list.

ALGORITHM:

Step 1: Start the process
Step 2: Create a arrays of some fixed size.
Step 3: Take two variables i and j, which will be at the 0th position of the arrays.
Step 4: Enter the number of elements and the elements.
Step 5: Elements will be sorted in ascending order using for loop.
Step 6: Run the process
Step 7: Stop the process

PROGRAM 4:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[20],n,temp,i,j;
clrscr();
printf("\n\n\t ENTER THE NUMBER OF TERMS:");
scanf("%d",&n);
printf("\n\t ENTER THE ELEMENTS OF THE ARRAY:");
for(i=0;i<n;i++)
{
gotoxy(25,11+i);
scanf("\n\t\t%d",&a[i]);
}
for(i=1;i<n;i++)
{
temp=a[i];
j=i-1;
while(temp<a[j]&&j>=0)
{
a[j+1]=a[j];
j=j-1;
}
a[j+1  ]=temp;
}
printf("\n\t THE ASCENDING ORDER LIST IS\n");
for(i=0;i<n;i++)
printf("\n\t\t\t%d",a[i]);
getch();
}
```

TEST CASE:

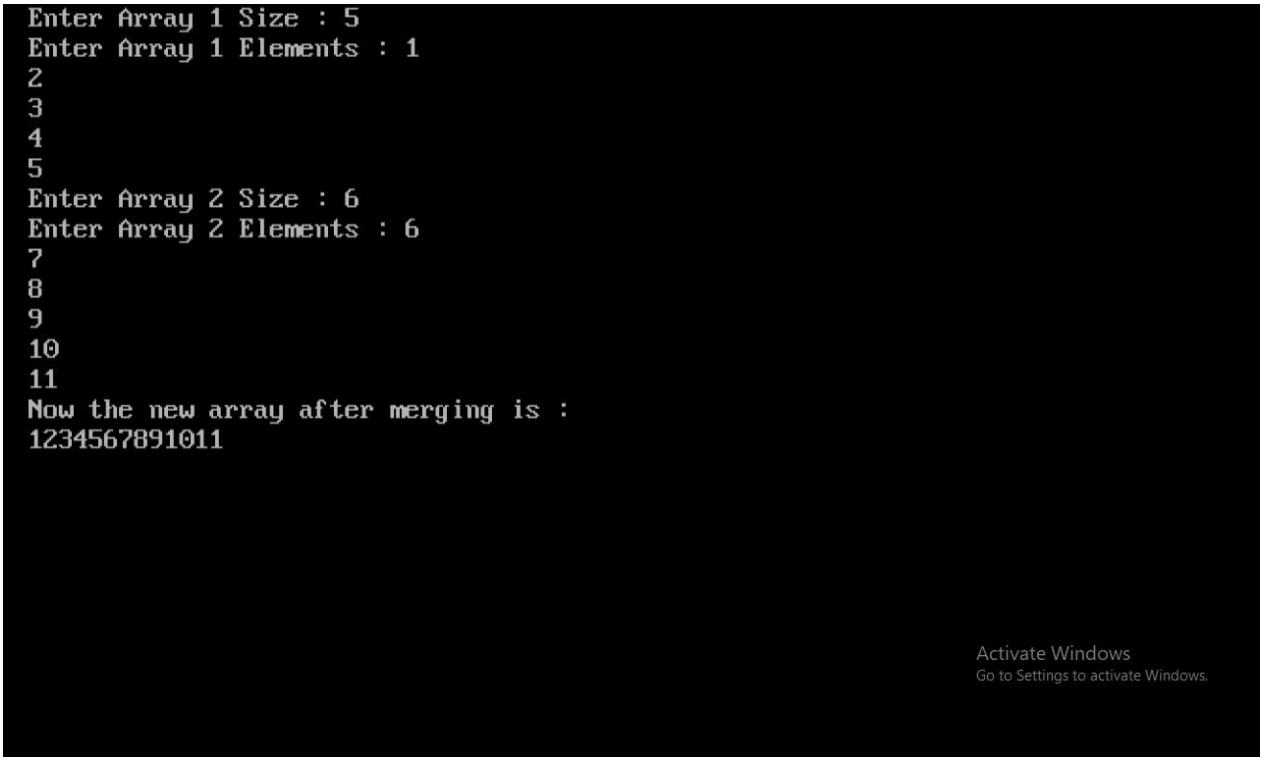
TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC01	CHECKING THEPROCESS FOR INPUT & OUTPUT	ADD THE HEADER FILE STATEMENT #INCLUDE<STDIO .H>	DISPLAY INPUT& OUTPUT STATEMENT S PROPERLY	DISPLAYE D INPUT & OUTPUT STATEMEN TS PROPERLY	SUCCESS
TC02	SUCCESSF UL PROCESS OFMAIN FUNCTION	MAIN FUNCTION STATEMENT SHOULD BE CHECKED	MAIN FUNCTION IS ALLOWING THE PROGRAM TO PROCESS	MAIN FUNCTION ALLOWED THE PROGRAM TO PROCESS	SUCCESS
TC03	INITIATION OF VARIABLES TOGET INPUT VALUES FOR ARRAY	INITIATE THE VARIABLES WITHTHE DATA TYPE INTEGER	ACCEPTAN CEOF INTEGER VARIABLE VALUES	NOT ACCEPT ED INTEGER VARIAB LE VALUES	FAILURE
TC04	GETTING INPUT VALUES FROM THE USER TO MENTION FIRST ARRAY SIZE	ENTER INPUT VALUEFOR ARRAY	ACCEPT THE INPUT FROMTHE USER	NOT ACCEPTE DTHE INPUT FROM THE USER	FAILURE
TC05	GIVE THE INPUTVALUES FOR FIRST ARRAY	ENTER THE FIRSTARRAY ELEMENTONE BY ONE	ACCEPT THE ARRAY ELEMENT FROM THE USER	ACCEPTE D THE ARRAY ELEMEN T FROM THE USER	SUCCESS
TC06	GIVE THE INPUTVALUES FOR SECOND ARRAY	ENTER THE SECONDARRAY ELEMENT ONE BY ONE	ACCEPT THE ARRAY ELEMENT FROM THE USER	ACCEPTE D THE ARRAY ELEMEN T FROM THE USER	SUCCESS
TC07	PROCESS FOR MERGING OFTWO ARRAYS	USE FOR LOOP FORMERGING	STORE MERGED FIRST ARRAY ELEMENT IN ARRAY VARIABLE MERGE[i]	STORED MERGED FIRST ARRAY ELEMENT INARRAY VARIABLE MERGE[i]	SUCCESS

TC08	PROCESS FOR MERGING OFTWO ARRAYS	USE FOR LOOP FORMERGING	STORE MERGED SECOND ARRAY ELEMENT INARRAY VARIABL E MERGE[K]	STORE MERGED SECOND ARRAY ELEMENT INARRAY VARIABL E MERGE[K]	SUCCESS
TC09	IDENTIFY THETOTAL ARRAY SIZE	ADD TWO ARRAY ELEMENT TO IDENTIFY THE TOTAL ARRAY SIZE	STORE THE ARRAY SIZE IN THE VARIABLE SIZE	STORED THE ARRAY SIZE IN THE VARIABLE SIZE	SUCCESS
TC10	PRINT THE MERGED AND SORTED ARRAY ELEMENT	USE FOR LOOP TO PRINT THE MERGEDARRAY ELEMENT ONE BY ONE	PRINT THE MERGED AND SORTED ARRAY ELEMENT ONE BY ONE	PRINTED THE MERGED AND SORTED ARRAY ELEMENT ONE BY ONE	SUCCESS

ERROR CORRECTION:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC03	INITIATION OFVARIABLES TOGET INPUT VALUES FOR ARRAY	CHECK THE DATA TYPE OF INITIATED VARIABLES	ACCEPTANCE OF INTEGER VARIABLE VALUES	ACCEPTD INTEGER VARIABLE VALUES	SUCCESS
TC04	GETTING INPUTVALUES FROM THE USER TO MENTION FIRST ARRAY SIZE	CHECK THE ARRAYSIZE AND INPUT VALUE OF ARRAY	ACCEPT THE INPUT FROM THE USER	ACCEPTE THE INPUT FROM THE USER	SUCCESS

OUTPUT :



RESULT:

EX.NO: 05 **/*STACK OPERATION*/**

DATE:

AIM:

To Test the C program: Experiment the operations of a stack using array implementation.

ALGORITHM:

Step 1:Start the process
Step 2:Get the number of element from user.
Step 3:And get the elements using array.
Step 4:Do the stack operation using push(),pop(),display() functions.
Step 4:Get the choice of operation from user using switch case.
Step 5:Run the process and display the result.
Step 6:Stop the process

PROGRAM 5:

```
#include<stdio.h>
int stack[100],choice,n,top,x,i;
void push(void);
void pop(void);
void display(void);
int main()
{
clrscr();
top=-1;
printf("\n Enter the size of STACK[MAX=100]:");
scanf("%d",&n);
printf("\n\t STACK OPERATIONS USING ARRAY");
printf("\n\t_____");
printf("\n\t 1.PUSH\n\t 2.POP\n\t 3.DISPLAY\n\t 4.EXIT");
do
{
printf("\n Enter the Choice:");
scanf("%d",&choice);
switch(choice)
{
case 1:
{
push();
break;
}
case 2:
{
pop();
break;
}
case 3:
{
display();
break;
}
case 4:
{
printf("\n\t EXIT POINT ");
break;
}
default:
```

```
{
printf ("\n\t Please Enter a Valid Choice(1/2/3/4)");
}}}
while(choice!=4);
return 0;
}
void push()
{
if(top>=n-1)
{
printf("\n\tSTACK is over flow");
}
else
{
printf(" Enter a value to be pushed:");
scanf("%d",&x);
top++;
stack[top]=x;
}
}
void pop()
{
if(top<=-1)
{
printf("\n\t Stack is under flow");
}
else
{
printf("\n\t The popped elements is %d",stack[top]);
top--;
}
}
void display()
{
if(top>=0)
{
printf("\n The elements in STACK \n");
for(i=top; i>=0; i--)
printf("\n%d",stack[i]);
printf("\n Press Next Choice");
}
else
{
printf("\n The STACK is empty");
}}
```

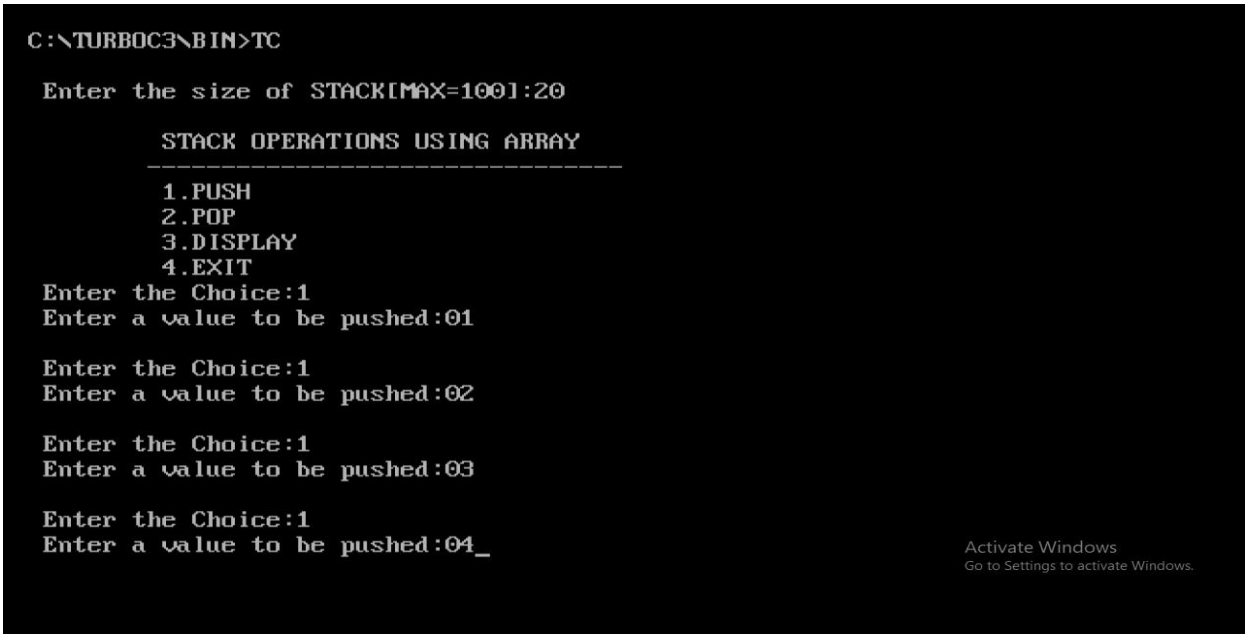
TEST CASE:

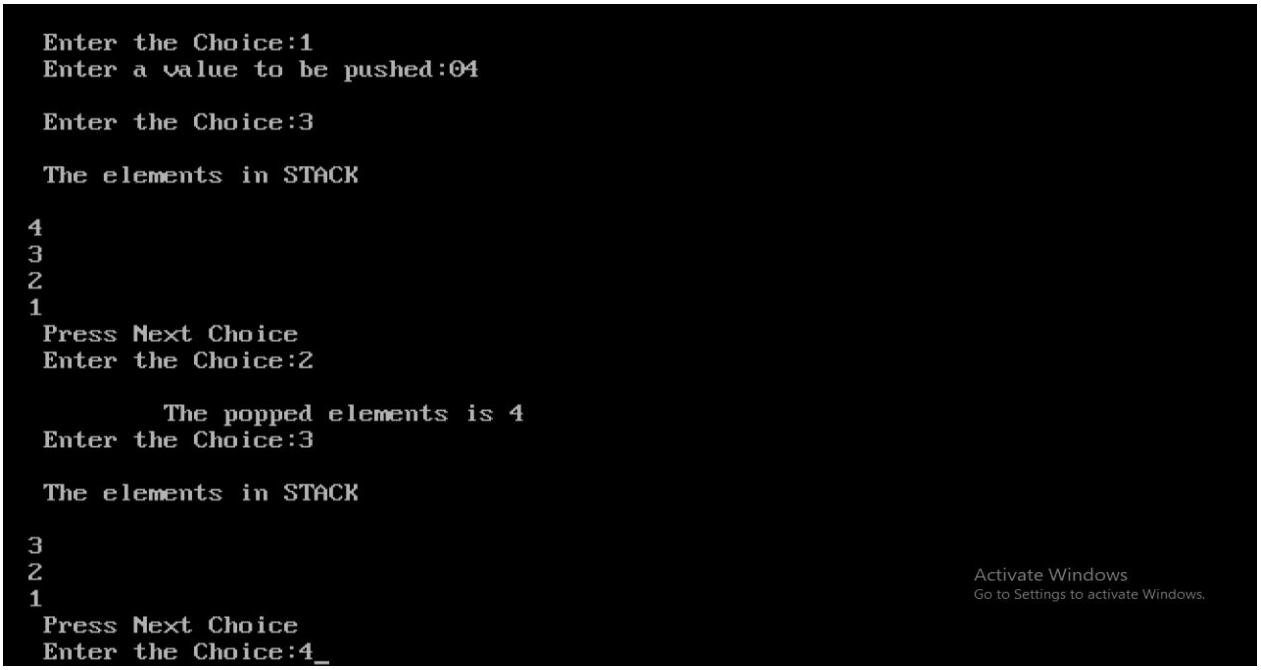
TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC-01	ACCEPTANCE OF<stdio.h> FILE	<stdio.h> HEADERFILE	ACCEPTING <stdio.h> HEADERFILE	ACCEPTED <stdio.h> HEADER FILE	SUCCESS
TC-02	ACCEPTANCE OF void push(void) PROTOTYPE	void push(void) PROTOTYPE	ACCEPTING voidpush(void) PROTOTYPE	ACCEPTED voidpush(void) PROTOTYPE	SUCCESS
TC-03	ACCEPTING THE DECLARATION OF choice VARIABLE	choice VARIABLE	THE VARIABLE choice SHOULD BE ACCEPTED	THE VARIABLE choice IS NOT ACCEPTED	FAILURE
TC-04	ACCEPTANCE OF DECLARATION OF top VARIABLE	top VARIABLE	THE VARIABLE topSHOULD BE ACCEPTED	THE VARIABLE topIS ACCEPTED	SUCCESS
TC-05	ACCEPTANCE OF printf STATEMENTTO GET THE SIZE OF STACK	printf STATEMENT	ACCEPTING THE printf STATEMENT	printf STATEMENT IS ACCEPTED	SUCCESS
TC-06	ACCEPTANCE OF scanf STATEMENTTO GET THE SIZE OF STACK	scanf STATEMENT	NUMBER WITH DECIMAL POINTSHOULD NOT BE ACCEPTED	NUMBER WITH DECIMAL POINT IS ACCEPTED	FAILURE
TC-07	ACCEPTANCE OF scanf STATEMENTTO GET THEchoice	scanf STATEMENT	NUMBER WITH DECIMAL POINTHOULD NOTACCEPTED	NUMBER WITH DECIMAL POINT ISNOT ACCEPTED	SUCCESS
TC-08	CHECKING THE SYNTAX OF switch STATEMENT	switch STATEM ENT	ACCEPTING THE switch STATEMEN T	switch STATEMENT IS ACCEPTED	SUCCESS
TC-09	ACCEPTING return STATEMENT	return STATEM ENT	ACCEPTING return STATEMENT	return STATEMENT IS ACCEPTED	SUCCESS
TC-10	ACCEPTIN G if CONDITIO N	if CONDITION	ACCEPTING THE if CONDITION	if CONDITION IS ACCEPTED	SUCCESS

ERROR CORRECTION:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC-03	ACCEPTING THE DECLARATION OF choice VARIABLE	choice VARIABLE	THE VARIABLE choice SHOULD BE ACCEPTED	THE VARIABLE choice IS ACCEPTED	SUCCESS
TC-06	ACCEPTANCE OF scanf STATEMENT TO GET THE SIZE OF STACK	scanf STATEMENT	NUMBER WITH DECIMAL POINT SHOULD NOT BEACCEPTED	NUMBER WITH DECIMAL POINT ISNOT ACCEPTED	SUCCESS

OUTPUT :





RESULT:

EX.NO: 06 **/*QUEUE OPERATION*/**

DATE:

AIM:

To Test the C program: Menu-driven option for queue operations like add, remove and display.

ALGORITHM:

Step 1:Start the process
Step 2:Get the element from user using array
Step 3:Do the queue operation using insert(),del(),display() functions.
Step 4:Get the choice of operation from user using switch case.
Step 5:Run the process and display the result.
Step 6:Stop the process

PROGRAM 6:

```
#include<stdio.h>
#include<stdlib.h>
#define max_size 5
int queue[max_size],front=-1,rear=-1;
void insert();
void del();
void display();
int main()
{
int choice;
do
{
printf("\n\n--------QUEUE OPERATIONS ----------\n");
printf("1.Insert\n");
printf("2.Delete\n");
printf("3.Display\n");
printf("4.Exit\n");
printf("---------------------- ");
printf("\nEnter your choice:\t");
scanf("%d",&choice);
switch(choice)
{
case 1:
insert();
break;
case 2:
del();
break;
case 3:
display();
break;
case 4:
exit(0);
break;
default:
printf("\nInvalid choice:\n");
break;
}
}
while(choice!=4);
return 0;
```

```
}
void insert()
{
int item;
if(rear==(max_size-1))
{
printf("\nQueue Overflow:");
}
else
{
printf("Enter the element to be inserted:\t");
scanf("%d",&item);
rear=rear+1;
queue[rear]=item;
if(front==-1)
front=0;
}
}
void del()
{
int item;
if(front==-1)
{
printf("\nQueue Underflow:");
}
else
{
item=queue[front];
printf("\nThe deleted element: %d\t",item);
if(front==rear)
{
front=-1;
rear=-1;
}
else
{
front=front+1;
}
}
}
void display()
{
int i;
```

```
if(front==-1)
{
printf("\nQueue is Empty:");
}
else
{
printf("\nThe queue elements are:\n" );
for(i=front;i<=rear;i++)
{
printf("%d\t",queue[i]);
}
}
```

TEST CASE:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC-01	ACCEPTANCE OF<stdio.h> FILE	<stdio.h> HEADERFILE	ACCEPTING <stdio.h> HEADERFILE	ACCEPTED <stdio.h> HEADERFILE	SUCCESS
TC-02	ACCEPTANCE OF void insert() PROTOTYPE	void insert() PROTOTYPE	ACCEPTING voidinsert() PROTOTYPE	ACCEPTED voidinsert() PROTOTYPE	SUCCESS
TC-03	ACCEPTING THE DECLARATION OF choice VARIABLE	choice VARIABLE	THE VARIABLE choice SHOULDBE ACCEPTED	THE VARIABLE choice IS NOT ACCEPTED	FAILURE
TC-04	ACCEPTANCE OF do while STATEMENT	do while STATEMENT	ACCEPTING THE do while STATEMENT	do while STATEMENT IS ACCEPTED	SUCCESS
TC-05	ACCEPTANCE OF printf STATEMENTTO GET THE SIZE OF STACK	printf STATEMENT	ACCEPTING THE printf STATEMENT	printf STATEMENT ISACCEPTED	SUCCESS
TC-06	ACCEPTANCE OF scanf STATEMENTTO GET THE choice	choice STATEMENT	NUMBER WITH DECIMAL POINT SHOULD NOT BE ACCEPTED	NUMBER WITH DECIMAL POINT IS ACCEPTED	FAILURE
TC-07	CHECKING THE SYNTAX OF switch STATEMENT	switch STATEMENT	ACCEPTING THE switch STATEMENT	switch STATEMENT IS ACCEPTED	SUCCESS
TC-08	ACCEPTANCE OF case STATEMENTTO GET THE choice	case STATEMENT	ACCEPTING THE case STATEMENT	case STATEMENTIS ACCEPTED	SUCCESS
TC-09	ACCEPTING return STATEMENT	return STATEMENT	ACCEPTING return STATEMENT	return STATEMENT IS ACCEPTED	SUCCESS
TC-10	ACCEPTING if CONDITION	if CONDITION	ACCEPTING THE if CONDITION	if CONDITION ISACCEPTED	SUCCESS

ERROR CORRECTION:

TC-03	ACCEPTING THE DECLARATION OF choice VARIABLE	choice VARIABLE	THE VARIABLE choice SHOULD BEACCEPTED	THE VARIABLE choice IS ACCEPT ED	SUCCESS
TC-06	ACCEPTANCE OF scanf STATEMENT TO GET THE choice	choice STATEME NT	NUMBER WITH DECIMAL POINT SHOULD NOT BE ACCEPTED	NUMBER WITH DECIMAL POINT ISNOT ACCEPTED	SUCCESS

OUTPUT :

```
C:\TURBOC3\BIN>TC


--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
------------------------
Enter your choice:      1
Enter the element to be inserted:       20


--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
------------------------
Enter your choice:      1
Enter the element to be inserted:       30_

2.Delete
3.Display
4.Exit
------------------------
Enter your choice:      1
Enter the element to be inserted:       40


--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
------------------------
Enter your choice:      1
Enter the element to be inserted:       50


--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
------------------------
Enter your choice:      3
```

```
3.Display
4.Exit
-----------------------
Enter your choice:      3

The queue elements are:
20      30      40      50

--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
-----------------------
Enter your choice:      2

The deleted element: 20

--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
-----------------------
Enter your choice:      3_
```



```
3.Display
4.Exit
-----------------------
Enter your choice:      2

The deleted element: 20

--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
-----------------------
Enter your choice:      3

The queue elements are:
30      40      50

--------QUEUE OPERATIONS-----------
1.Insert
2.Delete
3.Display
4.Exit
-----------------------
Enter your choice:      4_
```


RESULT:

EX.NO: 07 **/*PALINDROME*/**

DATE:

AIM:

To Test the C++ program: Palindrome string checking program (using pointers)

ALGORITHM:

Step 1:Start the process
Step 2:Get the String from the user
Step 3:Hold the string and copy of the string in two different variables
Step 4:Reverse the string in one variable
Step 5:Compare the two variables
Step 6:If both are same Print String is a Palindrome
Step 7:Else Print String is not a palindrome
Step 8:Stop the process

PROGRAM 7:

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
void main()
{
char *s1,*s2,ch;
clrscr();
cout<<"\n PALINDOROME";
gets(s1);
s2=new char (strlen(s1));
strcpy(s2,s1);
strrev(s2);
if(strcmp(s1,s2))
cout<<"\n THE GIVEN STRING IS NOT A PALINDROME:";
else
cout<<"\n THE GIVEN STRING IS A PALINDROME:";
cout<<"\n DO YOU WANT TO CONTINUE TYPE(Y OR N):";
cin>>ch;
if(ch=='y'||ch=='n')
getch();
}
```

TEST CASE:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC01	CHECKING THEPROCESS FOR INPUT & OUTPUT	ADD THE HEADER FILE STATEMENT #INCLUDE<IOSTREAM.H>	DISPLAY INPUT & OUTPUT STATEMENTS PROPERLY	DISPLAYED INPUT & OUTPUT STATEMENTS PROPERLY	SUCCESS
TC02	SUCCESSFUL PROCESS OF MAIN FUNCTION	MAIN FUNCTION STATEMENT SHOULD BE CHECKED	MAIN FUNCTION IS ALLOWING THE PROGRAM TO PROCESS	MAIN FUNCTION ALLOWED THE ROGRAM TO PROCESS	SUCCESS
TC03	INITIATION OF VARIABLE NAME STRING	INITIATE THE VARIAVBLE STRING WITH THE DATA TYPE CHARACTER ARRAY	ACCEPTANCE OF CHARACTER ARRAY VALUES	NOT ACCEPTANCE OF CHARACTER ARRAY VARIABLE VALUES	FAILURE
TC04	GETTING INPUT VALUES FROM THE USER	ENTER INPUT TO FIND WHETHER THE GIVEN INPUT IS PALINDROME OR NOT	ACCEPT THE INPUT FROM THE USER	ACCEPTEDTHEINPUT FROM THE USER	SUCCESS
TC05	CHECK THE GIVEN INPUT WHETHER IT IS CORRECT OR NOT	ENTER THE INPUT VALUEAS COLLECTION OF CHARACTERS	ACCEPT AND DISPLAY THE GIVEN INPUT VALUE	NOT ACCEPTED AND DISPLAYTHE GIVEN INPUT VALUE	FAILURE
TC06	INITIATE THEPROCESS OF WHILE LOOP	PROCESSING OF ASSIGINING P1 VALUEINTO P3	ASSIGN ALL CHARACTERS FROM P1 TO P3	ASSIGNED ALL CHARACTERS FROM P1 TO P3	SUCCESS
TC07	INITIATE THE PROCESS OF IF STATEMENT PROCESS	GIVEN INPUT VALUES OFP1 AND P3 ARE EQUAL OR NOT	ASSIGN FLAG VALUE AS 1 IF IT IS EQUAL	ASSIGNED FLAG VALUEIS 1	SUCCESS
TC08	INITIATE THE PROCESS OF IF STATEMENT	GIVEN INPUT VALUES OFP1 AND P3 ARE EQUAL OR NOT	ASSIGN FLAG VALUE AS 0 IF IT IS NOT EQUAL	ASSIGNED FLAG VALUEIS 0	SUCCESS
TC09	INITIATE THE PROCESS OF IF STATEMENT	FIND THE GIVEN VALUE IS PALINDROME OR NOT	DISPLAY THE RESULT AS NOT A PALINDROME IF THE FLAG VALUE IS 0	DISPLAYED THE RESULT AS NOT A PALINDROME	SUCCESS
TC10	INITIATE THE PROCESS OF IF	FIND THE GIVEN VALUE IS PALINDROME OR NOT	DISPLAY THE RESULT AS	DISPLAYED THE RESULT	SUCCESS

	STATEMENT		PALINDROMEIF THE FLAG VALUE IS 1	AS PALINDROME	

ERROR CORRECTION:

TEST ID	TEST DESCRIPTION	TEST STEPS	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC03	INITIATION OF VARIABLE NAME SRTING	CHECK THE GIVENDATA TYPE CHARACTER IS CORRECT OR NOT	ACCEPTANCE OF CHARACTER VARIABLE VALUES	ACCEPTED CHARACTER VARIABLE VALUES	SUCCESS
TC05	CHECK THE GIVEN INPUT WHETHER IT IS CORRECT OR NOT	GIVE INPUT AS COLLECTION OFARRAY VALUES	ACCEPT AND DISPLAY THE GIVEN INPUT VALUE	ACCEPTED AND DISPLAYED THE GIVEN INPUT VALUE	SUCCESS

OUTPUT :

RESULT: