**SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA**

**(UNIVERSITY ESTABLISHED under section 3 of UGC Act 1956)**

**ENATHUR,** **KANCHIPURAM – 631 561**



**PROGRAMMING IN C LAB**

**LABORATORY RECORD**

**Name :**

**Reg. No :**

**Class :** I YEAR BCA

**Subject :** UDSF231P60 - PROGRAMMING IN C LAB

**SRI CHANDRASEKHARENDRA SARASWATHI**

**VISWA MAHAVIDYALAYA**

**(University Established under section 3 of UGC Act 1956)**

****

**BONAFIDE CERTIFICATE**

**This is to Certify that this is the bonafide record of work done by**

**Mr./Ms.** **, with Reg.No of I Year BCA in the Programming in C Lab during the year 2025.**

**Staff-in-charge** **Head of the Department**

**Submitted for the Practical Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner** **External Examiner.**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Date** | **Title** | **Page No.** | **Signature** |
| 1 | 17-07-24 | FIBONACCI SERIES | 1 |  |
| 2 | 24-07-24 | PRIME NUMBER OR NOT | 3 |  |
| 3 | 31-07-24 | [SQUARE AND CUBE OF NUMBERS](#squarecube) | 5 |  |
| 4 | 07-08-24 | [GENERATE ODD NUMBERS](#Odd) | 7 |  |
| 5 | 14-08-24 | REVERSE ORDER OF NUMBER | 9 |  |
| 6 | 21-08-24 | SORTING OF AN ARRAY | 11 |  |
| 7 | 04-09-24 | ADDITION OF TWO MATRICES | 14 |  |
| 8 | 16-09-24 | MULTIPLICATION OF TWO MATRICES | 17 |  |
| 9 | 25-09-24 | SORTING THE ARRAY OF NAMES | 20 |  |
| 10 | 23-10-24 | FACTORIAL OF AN INTEGER USING RECURSION | 23 |  |
| 11 | 30-10-24 | SWAPPING OF TWO VALUES USING FUNCTION | 25 |  |
| 12 | 06-11-24 | READ AND WRITE A FILE | 27 |  |

|  |  |  |
| --- | --- | --- |
| **1** | **SQUARE AND CUBE OF NUMBERS** | **DATE:** |

**AIM :**

To calculate the square and cube of 1 to n numbers

**ALGORITHM :**

**Step 1:** Start

**Step 2:** Declare integer variables limit( n) and counter (i)

**Step 3:** read the value n

**Step 4:** Use a **for loop** that runs from i = 1

**Step 5:** Calculate square = i \* i

**Step 6:** Calculate cube = i \* i \* i

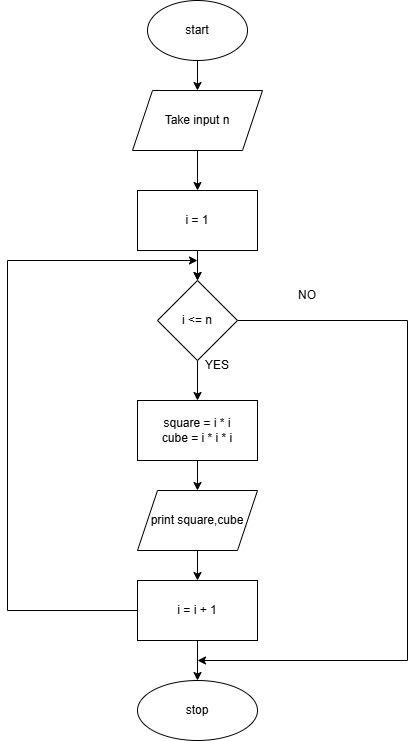
**Step 7:** Display i, square, and cube

**Step 8:** i = i + 1

**Step 9:** If i < n go to step 5 else goto step 10

**Step 10:** stop

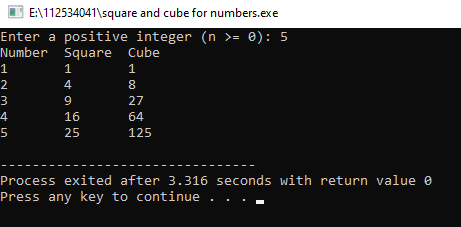
**FLOWCHART:**



**SOURCE CODE :**

<https://github.com/Rohith-Tech-king/c-lab/blob/main/square%20and%20cube%20for%20numbers.c>

**OUTPUT:**



**RESULT :**

Thus the program is compiled and executed successfully with verified output.

|  |  |  |
| --- | --- | --- |
| **2** | **GENERATE ODD NUMBERS** | **DATE:** |

**AIM :**

To display odd numbers from 1 to n

**ALGORITHM :**

**Step 1:** Start

**Step 2:** Declare integer variables limit (n) and counter (i)

**Step 3:** Read the value of n

**Step 4:** Use a **for loop** that runs from i = 1

**Step 5:** if i % 2 != 0 (i.e., i is odd) goto step 6, else goto step 7

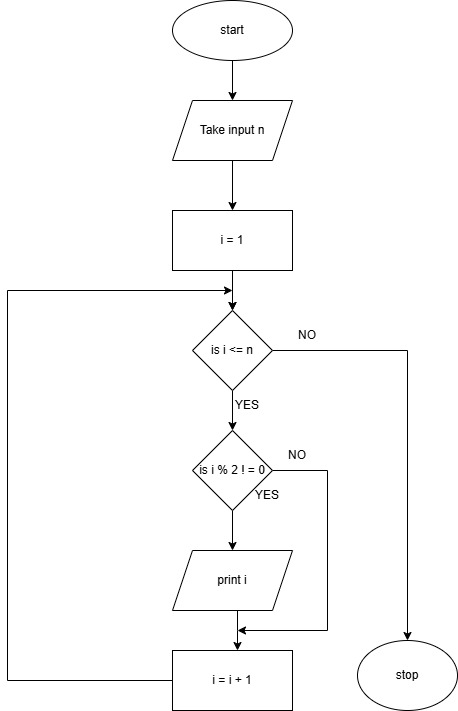
**Step 6:** Print i

**Step 7:** i=i+1

**Step 8:** Goto Step 5

**Step 9:** Stop

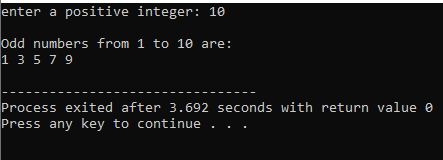
**FLOWCHART:**



**SOURCE CODE:**

[**https://github.com/Rohith-Tech-king/c-lab/blob/main/odd%20num.c**](https://github.com/Rohith-Tech-king/c-lab/blob/main/odd%20num.c)

**OUTPUT:**

****

**RESULT :**

Thus the program is compiled and executed successfully with verified output**.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 31-07-24 | 1. **CHECK NUMBER IS POSITIVE OR NEGATIVE OR ZERO** | 5 |

**AIM :**

**Program to check whether the given number is Positive or Negative or Zero using if…else if….**

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Declare the variable int num

**STEP 3 :** Read the value of num

**STEP 4 :** Print "Enter a Positive or Negative Number "

**STEP 5 :** Read the value of num

**STEP 6 :** if(num==0)

**STEP 7 :** Print "The given number num is Zero " num

**STEP 8 :** else if (num>0)

**STEP 9 :** Print "The given number num is Positive " num

**STEP 10 :** Else print "The given number num is Negative" num

**STEP 11 :** Stop

**SOURCE CODE :**

int num;

printf("enter the positive or negative number ");

scanf("%d",&num);

if(num==0)

{

printf("The given number %d is Zero ",num);

}

else if (num>0)

{

printf("The given number %d is Positive ",num);

}

else

{

printf("The given number %d is Negative",num);

}

return 0;

}

**OUTPUT:**

enter the positive or negative number 0

The given number 0 is Zero

enter the positive or negative number 10

The given number 10 is Postive

enter the positive or negative number -10

The given number -10 is Negative

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 07-08-24 | 1. **PALINDROME OR NOT** | 7 |

**AIM :**

**Program to find whether the give String is Palindrome or Not using for loop.**

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Declare the variable char str[30]

**STEP 3 :** Declare the variable int i,j,len

**STEP 4 :** Print "Enter a String:"

**STEP 5 :** Read the the String str

**STEP 6 :** len=strlen(str)

**STEP 7 :** for(i=0,j=len-1;i<len/2;i++,j--)

**STEP 8 :** if (str[i] != str[j])

**STEP 9 :** Print “Given String is Not a Palindrome” str

**STEP 10 :** exit(0)

**STEP 11 :** else

**STEP 12 :** Print “Given String is a Palindrome” str

**STEP 12 :** Stop

**SOURCE CODE :**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main()

{

char str[30];

int i,j,len;

printf("Enter a String:");

scanf("%s",str);

len=strlen(str);

for(i=0,j=len-1;i<len/2;i++,j--)

{

if (str[i] != str[j])

{

printf("%s is not a palindrome\n", str);

exit(0);

}

}

printf("%s is a palindrome\n", str);

return 0;

}

**OUTPUT:**

Enter a String:liril

liril is a palindrome

Enter a String:rajesh

rajesh is not a palindrome

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 14-08-24 | 1. **REVERSE ORDER OF NUMBER** | 9 |

**AIM :**

**Program to display the given Number ‘n’ in Reverse Order**

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Declare the variable int n,reverse=0,rem

**STEP 3 :** Print "enter a number:"

**STEP 4 :** Read the value of n

**STEP 5 :** while(n!=0)

**STEP 6 :** rem = n%10

**STEP 7 :** reverse = reverse \* 10 + rem

**STEP 8 :** n=n/10

**STEP 9 :** Print "Reversed number is: " reverse

**SOURCE CODE :**

#include<stdio.h>

int main()

{

int n,reverse=0,rem;

printf( "\nenter a number:");

scanf("%d",&n);

printf("\nReversed number is: ");

while(n!=0)

{

rem = n % 10;

printf("%d",rem);

n=n/10;

}

return 0;

}

**OUTPUT:**

enter a number: 123

Reversed number is:321

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 21-08-24 | **6. SORTING OF AN ARRAY** | 11 |

**AIM :**

**Program to Sort of the ‘n’ Numbers of Elements in a given Array**

**ALGORITHM :**

**STEP 1:** Start

**STEP 2:** Declare the variable int a[100], temp,i,j,n

**STEP 3:** Printf "enter the number of element in the array"

**STEP 4:** Read the value of n

**STEP 5:** for (i=0;i<n;i++)

**STEP 6:** Read the value of a[i]

**STEP 7:** Print "Before Sorting "

**STEP 8:** for(i=0; i<n; i++)

**STEP 9:** Print a[i]

**STEP 10:** for(i=0; i<n; i++)

**STEP 11:** for(j=i+1; j<n; j++)

**STEP 12:** if(a[i]>a[j])

**STEP 13:** temp = a[i]

**STEP 14:** a[i] = a[j]

**STEP 15:** a[j] = temp

**STEP 16:** Print "After Sorting "

**STEP 17:** for(i=0; i<n; i++)

**STEP 18:** Print a[i]

**STEP 19:** Stop

**SOURCE CODE :**

#include<stdio.h>

int main()

{

int a[100];

int temp,i,j,n;

printf("enter the number of element in the array");

scanf("%d",&n);

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

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

printf("Before Sorting ");

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

{

printf("%d ",a[i]);

}

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

{

for(j=i+1; j<n; j++)

{

if(a[i]>a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

printf("\nAfter Sorting ");

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

{

printf("%d ",a[i]);

}

return 0;

}

**OUTPUT:**

enter the number of element in the array 5

6

4

2

18

7

Before Sorting 6 4 2 18 7

After Sorting 2 4 6 7 18

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 04-09-24 | **7. ADDITION OF TWO MATRICES** | 14 |

**AIM :**

**Program to find the Addition of two Matrices using Two Dimensional Array**

**ALGORITHM :**

**STEP 1**: Start the Program.

**STEP 2** : Declare the MatrixA[10][10], MatrixB[10][10], MatrixC[10][10], row, col, i, j

**STEP 3** : Print “Enter the row and column of Matrix”

# **STEP 4** : Read the value of row and column.

# **STEP 5** : Print “Enter the elements of MatrixA”

# **STEP 6** : for(i=0; i<row; i++)

**STEP 7** : for(j=0; j<col; j++)

# **STEP 8** : Read the value of MatrixA[i][j]

**STEP 9** : Print “Enter the elements of MatrixB”

**STEP 10** : for(i=0; i<row; i++)

**STEP 11** : for(j=0; j<col; j++)

# **STEP 12** : Read the value of MatrixB[i][j]

**STEP 13** : for(i=0; i<row; i++)

**STEP 14** : for(j=0; j<col; j++)

# **STEP 15** : MatrixC[i][j]=MatrixA[i][j]+MatrixB[i][j]

**STEP 16** : Print “The Resultant MatrixC”

**STEP 17** : for(i=0; i<row; i++)

**STEP 18** : for(j=0; j<col; j++)

# **STEP 19** : Print “the Resultant Matrix “ MatrixC[i][j]

**STEP 20** : Stop.

**SOURCE CODE :**

#include<stdio.h>

void main()

{

int a[10][10],b[10][10],c[10][10], row, col, i, j;

printf("Enter the row and column:");

scanf("%d %d",&row, &col);

printf("Enter the elements of matrix A\n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

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

}

}

printf("Enter the elements of matrix B\n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

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

}

}

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

c[i][j] = a[i][j] + b[i][j];

}

}

printf("Resultant Matrix is \n");

for(i=0;i<row;i++)

{

for(j=0;j<col;j++)

{

printf(" %d",c[i][j]);

}

printf ("\n");

}

}

**OUTPUT:**

Output

Enter the row and column: 2 2

Enter the elements of matrix A

1

1

1

1

Enter the elements of matrix B

2

2

2

2

Resultant Matrix is

3 3

3 3

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 16-09-24 | **8. MULTIPLICATION OF TWO MATRICES** | 17 |

**AIM :**

**Program to find the Multiplication of two Matrices using Two Dimensional Array**

**ALGORITHM :**

**STEP 1**: Start the Program.

# **STEP 2**: Declare the MatrixA[10][10], MatrixB[10][10], MatrixC[10][10], row, col, m, n, i, j, k.

**Step 3:** Print “Enter the row and column of MatrixA”

**STEP 4**: Read the row and col

# **Step 5:** Print “Enter the row and column of MatrixB”

**STEP 6**: Read the row and col.

# **STEP 7**: if (col==row) then STEP 8 else STEP 3

# **STEP 8:** Print “Enter the Elements of MatrixA”

# **STEP 9**: for(i=0; i<row; i++)

**STEP 10:** for(j=0; j<col; j++)

**STEP 11:** Read the value of MatrixA[i][j]

# **STEP 12**: Print “Enter the Elements of MatrixB”

**STEP 13:** for(i=0; i<row; i++)

**STEP 14**: for(j=0; j<col; j++)

# **STEP 15**: Read the value of MatrixB[i][j]

**STEP 16:** for(i=0; i<row; i++)

**STEP 17**: for(j=0; j<col; j++)

**STEP 18**: MatrixC[i][j]=0

**STEP 19**: for(k=0; k<col; k++)

# **STEP 20**: MatrixC[i][j]= MatrixC[i][j]+MatrixA[i][k]+MatrixB[k][j]

**STEP 21**: Print “The Resultant MatrixC”

**STEP 22**: for(i=0; i<row; i++)

**STEP 23:** for(j=0; j<col; j++)

**STEP 24:** Print the elements of MatrixC[i][j]

**STEP 25:** Stop.

**SOURCE CODE :**

#include<stdio.h>

int main()

{

int a[10][10],b[10][10],c[10][10],row1,col1,row2,col2,i,j,k;

printf("enter the Matrix A row and column=");

scanf("%d%d",&row1,&col1);

printf("enter the Matrix B row and column=");

scanf("%d%d",&row2,&col2);

printf("enter the Matrix A elements:\n");

for(i=0;i<row1;i++)

{

for(j=0;j<col1;j++)

{

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

}

}

printf("enter the Matrix B elements:\n");

for(i=0;i<row2;i++)

{

for(j=0;j<col2;j++)

{

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

}

}

printf("The Resultant Matrix C:\n");

for(i=0;i<row1;i++)

{

for(j=0;j<col2;j++)

{

c[i][j]=0;

for(k=0;k<row2;k++)

{

c[i][j]=c[i][j] + a[i][k]\*b[k][j];

}

}

}

for(i=0;i<row1;i++)

{

for(j=0;j<col2;j++)

{

printf("%d\t",c[i][j]);

}

printf("\n");

}

return 0;

}

**OUTPUT:**

enter the Matrix A row and column=2

3

enter the Matrix B row and column=3

2

enter the Matrix A elements:

2

2

2

2

2

2

enter the Matrix B elements:

3

3

3

3

3

3

The Resultant Matrix C:

18 18

18 18

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 25-09-24 | **9. SORTING THE ARRAY OF NAMES** | 20 |

**AIM :**

**Program to Sort the Array of Names using for loop.**

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Declare variables int i,j,n

**STEP 3 :** Declare character array str[100][100], tempstr [100]

**STEP 4 :** Read the number of names

**STEP 6 :** for(i=0;i<n;i++)

**STEP 7 :** Read the number of names

**STEP 8 :** for(i=0;i<n;i++)

**STEP 9 :** for(j=i+1;j<n;j++)

**STEP 10 :** if(strcmp(str[i],str[j])>0)

**STEP 11 :** strcpy(s,str[i])

**STEP 12 :** strcpy(str[i],str[j])

**STEP 13 :** strcpy(str[j],s)

**STEP 14:** Stop

**SOURCE CODE :**

#include<stdio.h>

#include<string.h>

void main()

{

int i,j,n;

char str[10][30], tempstr[30];

printf("Enter the n number of String :");

scanf("%d",&n);

printf("Enter the String name :");

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

{

scanf("%s",str[i]);

}

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

{

for( j = i+1; j < n; j ++)

{

if(strcmp(str[i],str[j])>0)

{

strcpy(tempstr,str[i]);

strcpy(str[i],str[j]);

strcpy(str[j],tempstr);

}

}

}

printf("\nAfter Sorting the names are:\n");

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

{

printf("%s\n",str[i]);

}

}

**OUTPUT:**

Enter the n number of String :5

Enter the String name :

Ramesh

Abinav

Saravanan

Kamesh

Bala

After Sorting the names are:

Abinav

Bala

Kamesh

Ramesh

Saravanan

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 23-10-24 | **10. FACTORIAL OF AN INTEGER USING RECURSION** | 23 |

**AIM :**

**Program to find the Factorial of an Integer using Recursion**

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Declare the variable x,n as integer

**STEP 3 :** Read the value of n

**STEP 4 :** Call the function x=fact(n)

**STEP 5 :** Print the factorial value n,x

**STEP 6 :** Display factorial

**STEP 7:** Stop

Function:

**STEP 1:** Define the function int fact(int n)

**STEP 2:** if(n==0)

**STEP 3:** return 1

**STEP 4:** else

**STEP 5:** return(n\*fact(n-1))

**SOURCE CODE :**

#include<stdio.h>

int fact(int n)

{

if(n==0)

{

return(1);

}

else

{

return(n\*fact(n-1));

}

}

int main()

{

int res,val;

printf("Enter the number to find factorial :");

scanf("%d",&val);

res=fact(val);

printf("Factorial of the given number is %d", res);

return 0;

}

**OUTPUT :**

Enter the number to find factorial : 5

Factorial of 5 is 120

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 30-10-24 | **11. SWAPPING OF TWO VALUES USING FUNCTION** | 25 |

**AIM :**

**PROGRAM TO SWAP THE TWO VALUES USING FUNCTION**

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Declare two variables x,y as integer

**STEP 3** : Print “before swapping” x,y

**STEP 4** : Call swap(&x,&y)

**STEP 5 :** Print “after swapping” x,y

**STEP 6 :** Stop

function

**STEP 1 :** void swap(int\*a, int\*b)

**STEP 2 :** temp=\*a

**STEP 3 :** \*a=\*b

**STEP 4 :** \*b=temp

**SOURCE CODE :**

#include<stdio.h>

void swap(int a,int b)

{

int temp;

temp = a;

a =b;

b=temp;

printf("\n After swapping: %d %d",a,b);

}

int main()

{

int x,y;

printf("Enter the Value of X and Y :");

scanf("%d %d",&x,&y);

printf("Before swapping:%d %d",x,y);

swap(x,y);

return 0;

}

Output

Enter the Value of X and Y :10 20

Before swapping:10 20

After swapping: 20 10

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**

|  |  |  |
| --- | --- | --- |
| **DATE:** 06-11-24 | **12. READ AND WRITE A FILE** | 27 |

**AIM :**

**Program to perform Read and Write operation in a File**

**ALGORITHM :**

**STEP 1 :** Start

**STEP 2 :** Declare the file pointer FILE \*fp

**STEP 3 :** Declare the variable char ch

**STEP 4 :** Open the file in Write mode fp = fopen("hello.txt", "w")

**STEP 5 :** Print "Enter data and press enter"

**STEP 6 :** while( (ch = getchar()) != '\n')

**STEP 7 :** Write in the file by putc(ch,fp);

**STEP 8 :** Close the filefclose(fp);

**STEP 9 :** Open the file in Read mode fp = fopen("hello.txt", "r")

**STEP 10 :** Print The content read from the file is:

**STEP 11 :** Store file content while( (ch = getc(fp))!= EOF)

**STEP 12 :** Print the file content ch

**STEP 13** : Close the file

**STEP 14 :** Stop

**SOURCE CODE :**

#include<stdio.h>

void main()

{

FILE \*fp;

char ch;

fp = fopen("hello.txt", "w");

printf("Enter a line and press Enter:");

while( (ch = getchar()) != '\n')

{

putc(ch,fp);

}

fclose(fp);

fp = fopen("hello.txt", "r");

printf("The content read from the file is:\n");

while( (ch = getc(fp))!= EOF)

{

printf("%c",ch);

}

}

**OUTPUT:**

Enter a line and press Enter: WELCOME TO SCSVMV UNIVERSITY

The content read from the file is:

WELCOME TO SCSVMV UNIVERSITY

**RESULT :**

**Thus the program is compiled and executed successfully with verified output.**