

OOPS

NAME:SATYA NAGA RENU

REG:22BEC1456

Date:14.08.2023

Q1) Write a C program to perform addition and multiplication of two numbers using all four types of function definitions (Passing arguments and return value).

CODE:

```
#include <stdio.h>

int addition(int x,int y) {
    return x+y;
}

int multiplication(int x, int y) {
    return x * y;
}

int divison(int x,int y){
    return x/y;
}

int subtraction(int x,int y){
    return x-y;
}

int main() {
    printf("Satya Naga Renu\n");
    printf("22BEC1456\n");

    int num1, num2;

    printf("Enter two numbers: ");

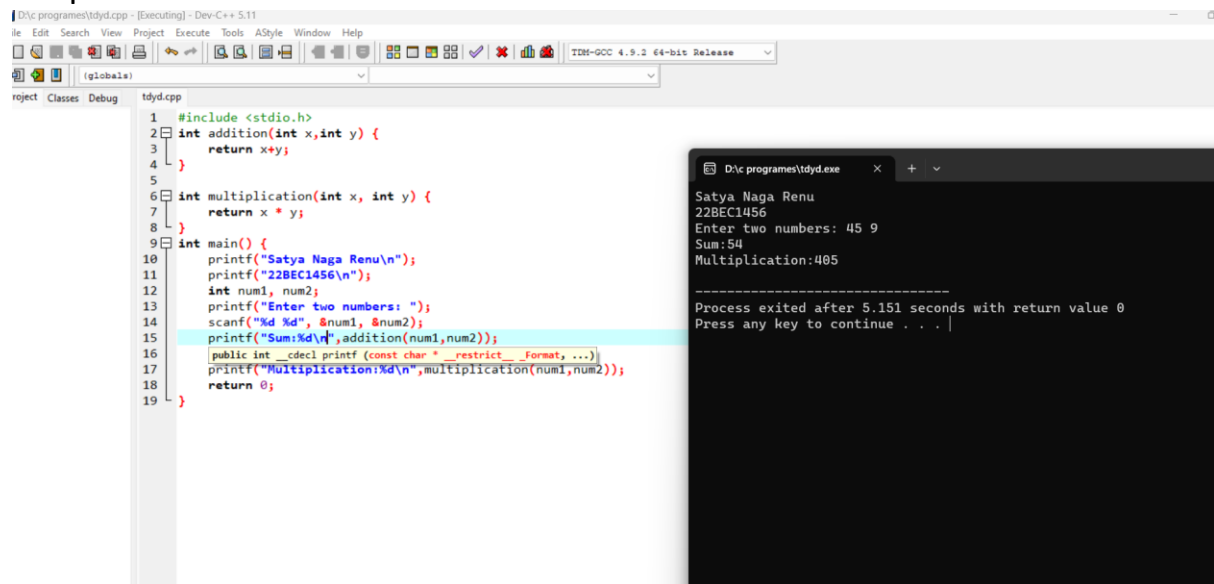
    scanf("%d %d", &num1, &num2);

    printf("Sum:%d\n",addition(num1,num2));

    printf("Multiplication:%d\n",multiplication(num1,num2));
```

}

Output:



The screenshot shows a C++ IDE with the file `tdyd.cpp` open. The code defines two functions: `addition` and `multiplication`, and a `main` function that prints the program name, a hexadecimal address, and prompts the user for two numbers. It then calls the `addition` and `multiplication` functions and prints their results. The output window shows the execution of the program, displaying the name 'Satya Naga Renu', the address '22BEC1456', and the results of the addition (45) and multiplication (405) of the input numbers 9 and 54. The program exits after 5.151 seconds with a return value of 0.

```
1 #include <stdio.h>
2 int addition(int x,int y) {
3     return x+y;
4 }
5
6 int multiplication(int x, int y) {
7     return x * y;
8 }
9
10 int main() {
11     printf("Satya Naga Renu\n");
12     printf("22BEC1456\n");
13     int num1, num2;
14     printf("Enter two numbers: ");
15     scanf("%d %d", &num1, &num2);
16     printf("Sum:%d\n",addition(num1,num2));
17     printf("Multiplication:%d\n",multiplication(num1,num2));
18     return 0;
19 }
```

Output:

```
Satya Naga Renu
22BEC1456
Enter two numbers: 45 9
Sum:54
Multiplication:405

Process exited after 5.151 seconds with return value 0
Press any key to continue . . .
```

Q2) Write a C program to convert a decimal number to binary using functions.

CODE:

```
#include <stdio.h>
```

```
void decimalToBinary(int decimal) {
```

```
    int binary[32];
```

```
    int index = 0;
```

```
    while (decimal > 0) {
```

```
        binary[index] = decimal % 2;
```

```
        decimal /= 2;
```

```
        index++;
```

```
    }
```

```
    printf("Binary representation: ");
```

```
    for (int i = index - 1; i >= 0; i--) {
```

```
        printf("%d", binary[i]);
```

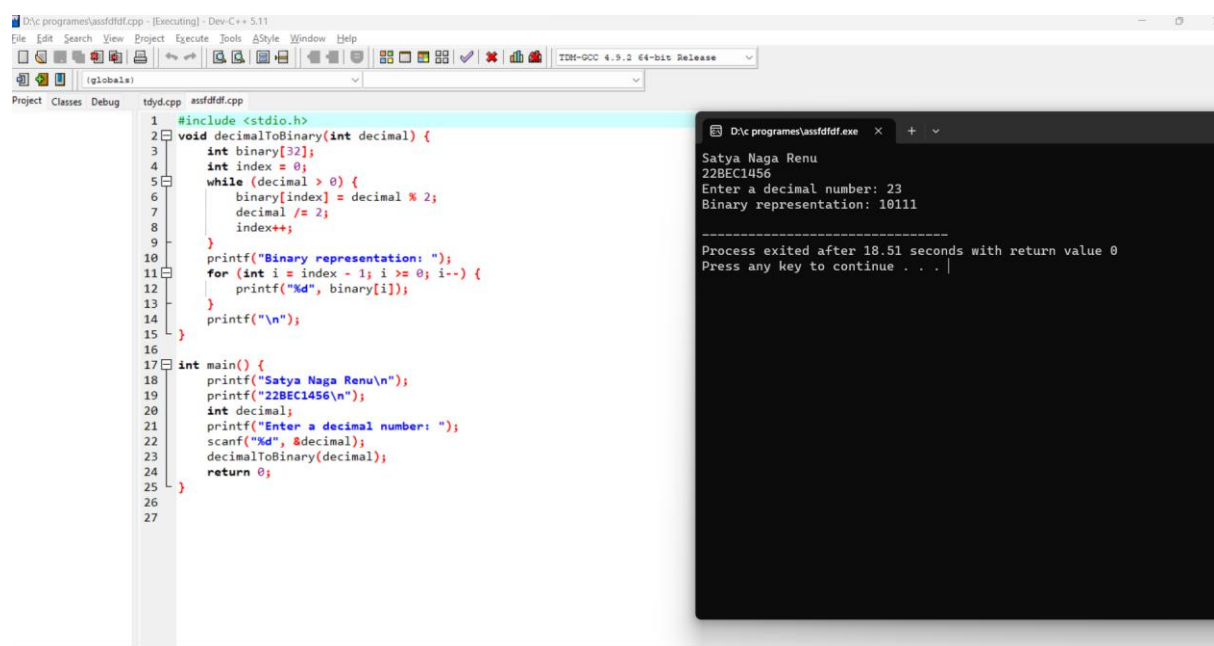
```
    }
```

```
    printf("\n");
```

```
}
```

```
int main() {  
    printf("Satya Naga Renu\n");  
    printf("22BEC1456\n");  
    int decimal;  
    printf("Enter a decimal number: ");  
    scanf("%d", &decimal);  
    decimalToBinary(decimal);  
    return 0;  
}
```

OUTPUT:



The screenshot displays a C++ IDE with two windows. The left window, titled 'assfddfd.cpp', shows the source code for a program that prints the name 'Satya Naga Renu' and the ID '22BEC1456', prompts for a decimal number, and converts it to binary. The right window, titled 'D:\c\programes\assfddfd.exe', shows the program's output: the name and ID are printed, the user enters '23', and the binary representation '10111' is displayed. The program then shows a message indicating it exited after 18.51 seconds and prompts the user to press any key to continue.

```
1 #include <stdio.h>  
2 void decimalToBinary(int decimal) {  
3     int binary[32];  
4     int index = 0;  
5     while (decimal > 0) {  
6         binary[index] = decimal % 2;  
7         decimal /= 2;  
8         index++;  
9     }  
10    printf("Binary representation: ");  
11    for (int i = index - 1; i >= 0; i--) {  
12        printf("%d", binary[i]);  
13    }  
14    printf("\n");  
15 }  
16  
17 int main() {  
18     printf("Satya Naga Renu\n");  
19     printf("22BEC1456\n");  
20     int decimal;  
21     printf("Enter a decimal number: ");  
22     scanf("%d", &decimal);  
23     decimalToBinary(decimal);  
24     return 0;  
25 }  
26  
27
```

```
Satya Naga Renu  
22BEC1456  
Enter a decimal number: 23  
Binary representation: 10111  
  
-----  
Process exited after 18.51 seconds with return value 0  
Press any key to continue . . .
```

Q3) Write a C program to insert an element into an array by passing an array to the function.

CODE: #include <stdio.h>

```
void insert(int arr[], int size, int index, int element) {
```

```

if (index < 0 || index > size) {
    printf("Invalid\n");
    return;
}

for (int i = size - 1; i >= index; i--) {
    arr[i + 1] = arr[i];
}

arr[index] = element;
}

int main() {
    int ar[100];
    int i;
    int size;
    printf("enter the size of array: ");
    scanf("%d",&size);
    printf("enter the elements in an array: ");
    for(i=0;i<size;i++){
        scanf("%d",&ar[i]);
    }
    printf("Satya Naga Renu\n");
    printf("22BEC1456\n");
    printf("Original array: ");
    for ( i = 0; i < size; i++) {
        printf("%d ", ar[i]);
    }
}

```

OUTPUT:

Q4) Write a C program to add two matrices by passing the two matrices to a function.

CODE:

```
#include <stdio.h>

void addMatrices(int mat1[][100], int mat2[][100], int result[][100], int rows,
int cols) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
}

int main() {
    int rows, cols;
    printf("satya naga renu\n");
    printf("22BEC1456\n");
    printf("Enter the number of rows and columns for the matrices: ");
    scanf("%d %d", &rows, &cols);

    int matrix1[100][100], matrix2[100][100], resultMatrix[100][100];
    printf("Enter elements of matrix 1:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &matrix1[i][j]);
        }
    }

    printf("Enter elements of matrix 2:\n");
```

```

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &matrix2[i][j]);
        }
    }

    addMatrices(matrix1, matrix2, resultMatrix, rows, cols);

    printf("Resultant matrix after addition:\n");

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", resultMatrix[i][j]);
        }

        printf("\n");
    }

    return 0;
}

```

OUTPUT:

```

1 #include <stdio.h>
2 void addMatrices(int mat1[][100], int mat2[][100], int result[][100], int rows, int cols) {
3     for (int i = 0; i < rows; i++) {
4         for (int j = 0; j < cols; j++) {
5             result[i][j] = mat1[i][j] + mat2[i][j];
6         }
7     }
8 }
9 int main() {
10     int rows, cols;
11     printf("satya naga renu\n");
12     printf("22BEC1456\n");
13     printf("Enter the number of rows and columns for the matrices: ");
14     scanf("%d %d", &rows, &cols);
15     int matrix1[100][100], matrix2[100][100], resultMatrix[100][100];
16     printf("Enter elements of matrix 1:\n");
17     for (int i = 0; i < rows; i++) {
18         for (int j = 0; j < cols; j++) {
19             scanf("%d", &matrix1[i][j]);
20         }
21     }
22
23     printf("Enter elements of matrix 2:\n");
24     for (int i = 0; i < rows; i++) {
25         for (int j = 0; j < cols; j++) {
26             scanf("%d", &matrix2[i][j]);
27         }
28     }
29     addMatrices(matrix1, matrix2, resultMatrix, rows, cols);
30     printf("Resultant matrix after addition:\n");
31     for (int i = 0; i < rows; i++) {
32         for (int j = 0; j < cols; j++) {
33             printf("%d ", resultMatrix[i][j]);
34         }
35         printf("\n");
36     }
37 }

```

```

satya naga renu
22BEC1456
Enter the number of rows and columns for the matrices: 3 3
Enter elements of matrix 1:
1 2 3 4 5 6 7 8 9
Enter elements of matrix 2:
9 8 7 6 5 4 3 2 1
Resultant matrix after addition:
10 10 10
10 10 10
10 10 10

-----
Process exited after 33.88 seconds with return value 0
Press any key to continue . . .

```

Date:19-08-2023

Q1) Write a C program using recursive functions to find the sum of numbers from 1 to n.

CODE:

```
#include <stdio.h>

int SUM(int n){
    if(n==1){
        return 1;
    }
    int sum=n;
    return sum+SUM(n-1);
}

int main(){
    printf("Satya Naga Renu\n");
    printf("22BEC1456\n");
    int a;
    printf("enter the number");
    scanf("%d",&a);
    printf("sum of the numbers %d",SUM(a));

}
```

OUTPUT:


```
1 #include <stdio.h>
2 int SUM(int n){
3     if(n==1){
4         return 1;
5     }
6     int sum=n;
7     return sum+SUM(n-1);
8 }
9 int main(){
10    printf("Satya Naga Renu\n");
11    printf("22BEC1456\n");
12    int a;
13    printf("enter the number");
14    scanf("%d",&a);
15    printf("sum of the numbers %d",SUM(a));
16 }
17 }
```

D:\c program\Untitled2.exe x + v

Satya Naga Renu
22BEC1456
enter the number10
sum of the numbers 55

Process exited after 31.28 seconds with return value 0
Press any key to continue . . . |

2. Write a C program to reverse a number using recursive function.

CODE:

```
#include <stdio.h>
```

```
int reverseNumber(int num, int reversed) {
```

```
    if (num == 0) {
        return reversed;
    }
```

```
    int digit = num % 10;
```

```
    reversed = reversed * 10 + digit;
```

```
    return reverseNumber(num / 10, reversed);
```

```
}
```

```
int main() {
```

```
    int num;
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &num);
```

```
    int reversed = reverseNumber(num, 0);
```

```
    printf("Reversed number: %d\n", reversed);
```

```

return 0;

}

```

OUTPUT:

The screenshot shows a C++ IDE with two windows. The left window, titled 'Untitled2.cpp', contains the following code:

```

1  #include <stdio.h>
2
3  int reverseNumber(int num, int reversed) {
4      if (num == 0) {
5          return reversed;
6      }
7
8      int digit = num % 10;
9      reversed = reversed * 10 + digit;
10     return reverseNumber(num / 10, reversed);
11 }
12
13 int main() {
14     int num;
15     printf("Enter a number: ");
16     scanf("%d", &num);
17
18     int reversed = reverseNumber(num, 0);
19     printf("Reversed number: %d\n", reversed);
20
21     return 0;
22 }

```

The right window, titled 'D:\c programes\Untitled2.exe', shows the program's output:

```

Enter a number: 1321
Reversed number: 1231

-----
Process exited after 4.565 seconds with return value 0
Press any key to continue . . .

```

3. Write a C program to find the sum of the digits of a number using recursive function.

CODE:

```
#include <stdio.h>
```

```
int reverseNumber(int num, int sum) {
```

```
    if (num == 0) {
```

```
        return sum;
```

```
    }
```

```
    int digit = num % 10;
```

```
    sum+= digit;
```

```
    return reverseNumber(num / 10, sum);
```

```
}
```

```
int main() {
```

```
    int num;
```

```
    printf("satya naga renu\n");
```

```

printf("22BEC1456\n");

printf("Enter a number: ");

scanf("%d", &num);


int reversed = reverseNumber(num, 0);

printf("Sum of digits in number: %d\n", reversed);


return 0;

}

```

OUTPUT:

```

1  #include <stdio.h>
2
3  int reverseNumber(int num, int sum) {
4      if (num == 0) {
5          return sum;
6      }
7      int digit = num % 10;
8      sum += digit;
9      return reverseNumber(num / 10, sum);
10 }
11
12 int main() {
13     int num;
14     printf("satya naga renu\n");
15     printf("22BEC1456\n");
16     printf("Enter a number: ");
17     scanf("%d", &num);
18
19     int reversed = reverseNumber(num, 0);
20     printf("Sum of digits in number: %d\n", reversed);
21
22     return 0;
23 }

```

```

D:\c programes\Untitled2.exe
satya naga renu
22BEC1456
Enter a number: 1234
Sum of digits in number: 10

-----
Process exited after 7.888 seconds with return value 0
Press any key to continue . . .

```

4. Write a C program to swap two numbers using pass by reference.

CODE:

```

#include <stdio.h>

int swap(int *c,int *d){

    int temp;

    temp=*c;

    *c=*d;

    *d=temp;
}

```

```

}

int main(){

    printf("Satya Naga Renu\n");

    printf("22BEC1456\n");

    printf("Enter the numbers");

    int a,b;

    scanf("%d%d",&a,&b);

    swap(&a,&b);

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

}

```

OUTPUT:

```

1 #include <stdio.h>
2 int swap(int *c,int *d){
3     int temp;
4     temp=*c;
5     *c=*d;
6     *d=temp;
7 }
9 int main(){
10     printf("Satya Naga Renu\n");
11     printf("22BEC1456\n");
12     printf("Enter the numbers");
13     int a,b;
14     scanf("%d%d",&a,&b);
15     swap(&a,&b);
16     printf("%d\n%d",a,b);
17 }

```

```

Satya Naga Renu
22BEC1456
Enter the numbers2 3
3
2
-----
Process exited after 5.551 seconds with return value 0
Press any key to continue . . .

```

5. Write a C program to add two numbers using pointers.

CODE: #include <stdio.h>

```

int sum(int *c,int *d){

    *c=*c+*d;

    return *c;

}

```

```

int main(){

    printf("Satya Naga Renu\n");

    printf("22BEC1456\n");

    printf("Enter the numbers");

    int a,b;

    scanf("%d%d",&a,&b);

    printf("%d\n%d",sum(&a,&b));

    return 0;

}

```

OUTPUT:

```

i.cpp  asfddf.cpp  Untitled2.cpp
#include <stdio.h>
int sum(int *c,int *d){
    *c=*c+*d;
    return *c;
}
int main(){
    printf("Satya Naga Renu\n");
    printf("22BEC1456\n");
    printf("Enter the numbers");
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d\n%d",sum(&a,&b));
    return 0;
}

```

```

D:\c programes\Untitled2.exe x + v
Satya Naga Renu
22BEC1456
Enter the numbers2 3
5
0
-----
Process exited after 4.301 seconds with return value 0
Press any key to continue . . .

```

6. Write a C program to find the length of the string and print the string using pointers.

CODE:

```

#include <stdio.h>

int main() {

    printf("satya naga renu\n");

    printf("22BEC1456\n");

    char str[100];

    printf("Enter a string: ");

    scanf("%s", str);

```

```

        int len=0;

char *ptr=str;
while (*ptr != '\0') {

    ptr++;

    len++;

}

printf("Length of the string is : %d",len);


return 0;

}

```

OUTPUT:

```

pp satya.cpp satya.cpp
#include <stdio.h>

int main() {
    printf("satya naga renu\n");
    printf("22BEC1456\n");
    char str[100];
    printf("Enter a string: ");
    scanf("%s", str);
    int len=0;
    char *ptr=str;
    while (*ptr != '\0') {
        ptr++;
        len++;
    }
    printf("Length of the string is : %d",len);

    return 0;
}

```

Date: 24-08.2023

1. Write a C program to calculate the sum of squares and sum of cubes of first n natural numbers. Get n as input from the user. Define two functions to compute the sum of squares and sum of cubes. Use function pointers to call the function.

CODE:

```
#include<stdio.h>

int sumsquare(int n){
    int sum1,i;
    for(i=1;i<=n;i++){
        sum1+=i*i;

    }
    return sum1;
}

int cube(int n){
    int sum2,j;
    for(j=1;j<=n;j++){
        sum2+=j*j*j;
    }
    return sum2;
}

int main(){
    int k;
    scanf("%d",&k);
    int(*ptr1)(int);
    ptr1=sumsquare;
    int(*ptr2)(int);
    ptr2=cube;
    printf("%d\n",(*ptr1)(k));
    printf("%d",(*ptr2)(k));
```

}

OUTPUT:

```
1  #include<stdio.h>
2  int sumsquare(int n){
3      int sum1,i;
4      for(i=1;i<=n;i++){
5          sum1+=i*i;
6      }
7      return sum1;
8  }
9
10 int cube(int n){
11     int sum2,j;
12     for(j=1;j<=n;j++){
13         sum2+=j*j*j;
14     }
15     return sum2;
16 }
17 int main(){
18     int k;
19     scanf("%d",&k);
20     int(*ptr1)(int);
21     ptr1=sumsquare;
22     int(*ptr2)(int);
23     ptr2=cube;
24     printf("%d\n",(*ptr1)(k));
25     printf("%d",(*ptr2)(k));
26
27
28
29
```