M. Ali. Arslan

19F-0348

**Assignment Of Recursion**

**Problem # 1**

#include<iostream>

using namespace std;

int fun(char arr[], int i)

{

int b = 0;

if (arr[i] == '\0')

return 0;

if (arr[i] == 'a' || arr[i] == 'A' || arr[i] == 'e' || arr[i] == 'E' || arr[i] == 'i' || arr[i] == 'I' || arr[i] == 'o' ||

arr[i] == 'O' || arr[i] == 'u' || arr[i] == 'U')

b = 1;

return b + fun(arr, i + 1);

}

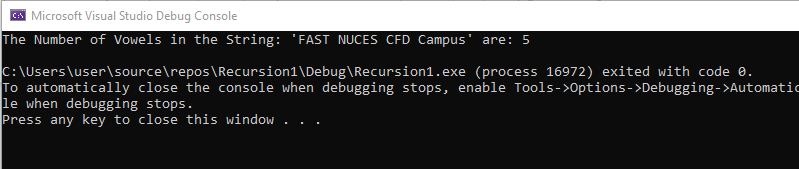
int main()

{

char str[] = "FAST NUCES CFD Campus";

cout << "The Number of Vowels in the String: 'FAST NUCES CFD Campus' are: " << fun(str, 0) << endl;

}



**Problem # 2**

#include<iostream>

using namespace std;

int sumofSq(int n)

{

if (n == 0)

{

return 0;

}

else if (n == 1)

{

return 1;

}

else

{

return (n \* n + sumofSq(n - 1));

}

}

int main()

{

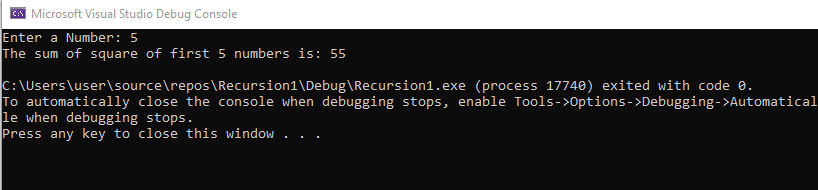
int num;

cout << "Enter a Number: ";

cin >> num;

cout << "The sum of square of first " << num << " numbers is: " << sumofSq(num) << endl;

}



**Problem #3**

#include<iostream>

using namespace std;

int sum(int arr[], int num)

{

if (num <= 0)

{

return 0;

}

else

{

return (sum(arr, num - 1) + arr[num - 1]);

}

}

int main()

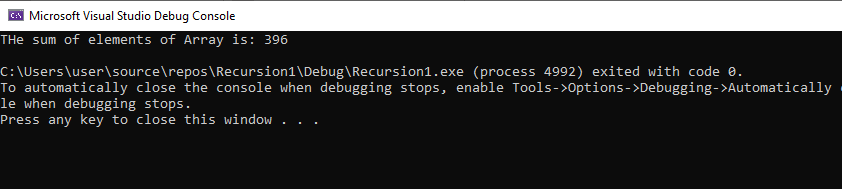
{

int array[] = { 8,245,65,22,53,3 };

int n = sizeof(array) / sizeof(array[0]);

cout << "THe sum of elements of Array is: " << sum(array, n) << endl;

}



**Problem # 4**

#include<iostream>

using namespace std;

bool check(char str[], int a, int b)

{

if (a == b)

{

return true;

}

if (str[a] != str[b])

{

return false;

}

if (a < b + 1)

{

return check(str, a + 1, b - 1);

return true;

}

}

bool ispal(char s[])

{

int n = strlen(s);

if (n == 0)

{

return true;

}

else

{

return check(s, 0, n - 1);

}

}

int main()

{

char str[100];

cout << "Enter a Word to check if it's Palindrome: ";

cin >> str;

if (ispal(str))

{

cout << str << " is Palindrome" << endl;

}

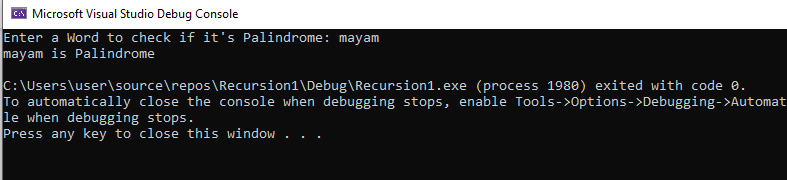
else

{

cout << str << " is not Palindrome" << endl;

}

}



**Problem # 5**

#include<iostream>

using namespace std;

int maximum(int array[], int num)

{

if (num == 1)

{

return array[0];

}

else

{

return max((array[num - 1]), maximum(array, num - 1));

}

}

int minimum(int a[], int length)

{

if (length == 1)

{

return a[0];

}

else

{

return min(a[length - 1], minimum(a, length - 1));

}

}

int main()

{

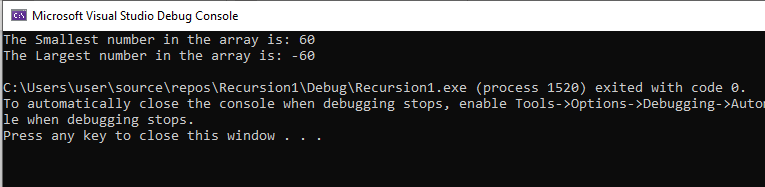
int arr[] = { 4,60,43,0,-60,34 };

int total = sizeof(arr) / sizeof(arr[0]);

cout << "The Smallest number in the array is: " << maximum(arr, total) << endl;

cout << "The Largest number in the array is: " << minimum(arr, total) << endl;

}



**Problem # 6**

#include<iostream>

using namespace std;

int check(int num)

{

int a = num / 10;

int b = 0;

if ((num % 10) == 0)

{

return 0;

}

else if ((num % 10) > (a % 10))

{

b = b + check(num / 10);

return b;

}

else

{

return 1;

}

}

int main()

{

int num, res;

cout << "Enter a number to check if its digits are in increasing order: ";

cin >> num;

res=check(num);

if (res == 0)

{

cout << num << " is in increasing order" << endl;

}

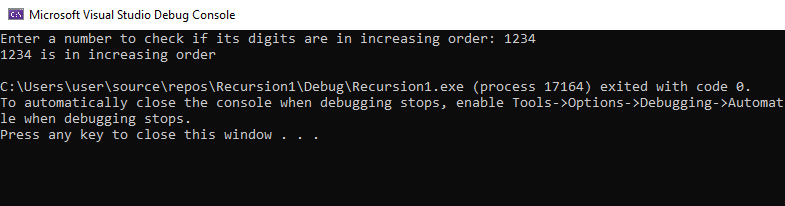
else

{

cout << num << " is not in increasing order" << endl;

}

}



**Problem # 7**

#include <iostream>

using namespace std;

int reverseDigits(int n)

{

static int temp = 0;

static int b = 1;

if (n > 0)

{

reverseDigits(n / 10);

temp = temp + (n % 10) \* b;

b = b \* 10;

}

return temp;

}

int main()

{

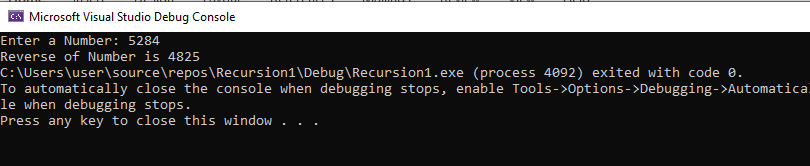
int num;

cout << "Enter a Number: ";

cin >> num;

cout << "Reverse of Number is " << reverseDigits(num);

}



**Problem # 8**

#include<iostream>

using namespace std;

int pow(int n, int p)

{

if (p == 0)

{

return 1;

}

else

{

return n \* pow(n, p - 1);

}

}

int main()

{

int n, power;

cout << "Enter a number: ";

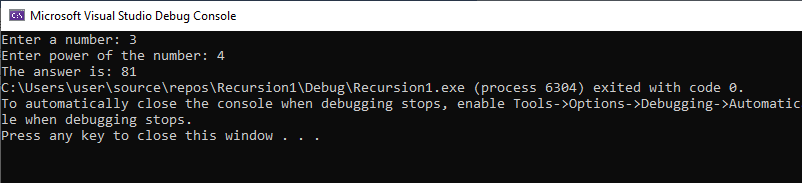
cin >> n;

cout << "Enter power of the number: ";

cin >> power;

cout << "The answer is: " << pow(n, power);

}



**Problem # 9**

#include <iostream>

using namespace std;

int gcd(int x, int y)

{

if (y != 0)

{

return gcd(y, x % y);

}

else

{

return x;

}

}

int main()

{

int x, y;

cout << "Enter Number 1: ";

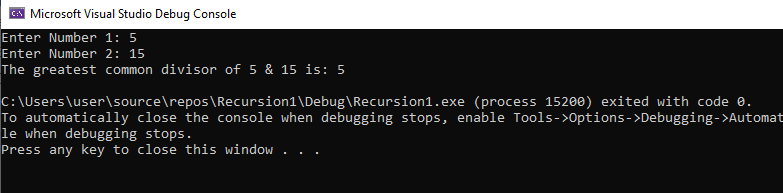
cin >> x;

cout << "Enter Number 2: ";

cin >> y;

cout << "The greatest common divisor of " << x << " & " << y << " is: "<< gcd(x, y) << endl;

}



**Problem # 10**

#include <iostream>

using namespace std;

int fun(int m, int n)

{

if (m == 0)

{

return n + 1;

}

else if ((m > 0) && (n == 0))

{

return fun(m - 1, 1);

}

else if ((m > 0) && (n > 0))

{

return fun(m - 1, fun(m, n - 1));

}

}

int main()

{

cout << "At m=3 & n=7, the Ackermann's function gives the value: " << fun(3, 7) << endl;

}

//At m=4, n=3 the program get crashed, throw an exception

