Basics Concepts of C++, Tokens Expression and Control structures

1. Write a C++ program that will output this passage by Deepak Chopra. Make sure your output looks exactly as shown here (including spacing, line breaks, punctuation, and the title and author). **Use cout and cin objects and endl manipulator.**

```
                    ******************************
*           Programmimg Assignment 1     *
                        *          Computer Programmimg I      *
                          *            Author : ???            *
*           Due Date: Thursday, Dec. 20    *
```

Question: Difference between \n and endl.

CODE:

```
#include<iostream>
using namespace std;
int main()
{

 cout<<"******************************"<<endl;
 cout<<"* Programming Assignment 1 *"<<endl;
 cout<<"* Computer Programming I *"<<endl;
 cout<<"*        Author : 18CE024  *"<<endl;
 cout<<"* Due Date:Thursday,Dec.20 *"<<endl;
 cout<<"******************************"<<endl;

 cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";

return 0;
}
```

OUTPUT:

```
******************************
*  Programming Assignment 1  *
*   Computer Programming I   *
*         Author : 18CE024   *
*  Due Date:Thursday,Dec.20  *
******************************

prepared by PRATIK DHORIYANI : 18CE024
```

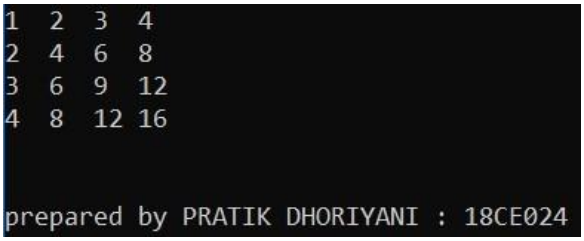
2. Write a program to create the following table. **Use endl and setw manipulator.**

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

CODE:

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
   int i,j;
   for(i=1;i<=4;i++)
   {
      for(j=1;j<=4;j++)
      {
         cout<<setw(3)<<left<<j*i;
      }
      cout<<"\n";
   }
   cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n";
   return 0;
}
```

OUTPUT:



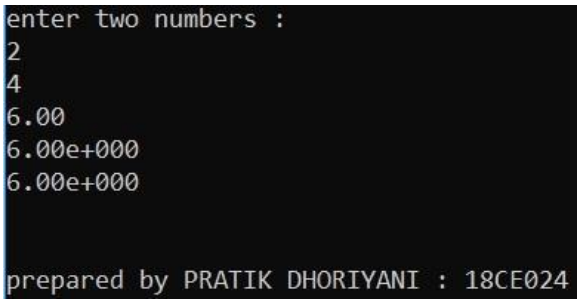
3. Write a C++ program to add two floating numbers using pointer. The result should contain only two digits after the decimal.**Use fixed, scientific and setprecision () manipulators for controlling the precision of floating point numbers.**

CODE:

```
#include<iostream>
#include<iomanip>
```

```
using namespace std;
int main()
{
  float *p,*q,a,b,sum;
  p=&a;
  q=&b;
  cout<<"enter two numbers : \n";
  cin>>*p>>*q;
  sum=(*p)+(*q);
  cout<<fixed<<setprecision(2)<<sum<<endl;
  cout<<scientific<<setprecision(2)<<sum<<endl;
  cout<<setprecision(2)<<sum<<endl;
  cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n";
  return 0;
}
```

OUTPUT:



4. Write a C++ program to find out sum of array element using Recursion.
Question: Show stepwise solution of **winding and unwinding phase of recursion**

CODE:

```
#include<iostream>
int add(int arr[],int n);
using namespace std;
int main()
{
   int n,sum;
   int arr[30];
   cout<<"enter the size of the array : ";
   cin>>n;
   cout<<"enter elements of the array : \n";
```

```
    for(int i=0;i<n;i++)
    {
       cout<<"arr["<<i<<"]= ";
       cin>>arr[i];
    }
    sum=add(arr,n);
    cout<<"total sum is "<<sum;
    cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n";
    return 0;
}
int add(int arr[],int n)
{
    if(n<=0)
    {
       return 0;
    }
    return add(arr,n-1) + arr[n-1];
}
```

OUTPUT:

```
enter the size of the array : 4
enter elements of the array :
arr[0]= 1
arr[1]= 2
arr[2]= 3
arr[3]= 4
total sum is 10

prepared by PRATIK DHORIYANI : 18CE024
```

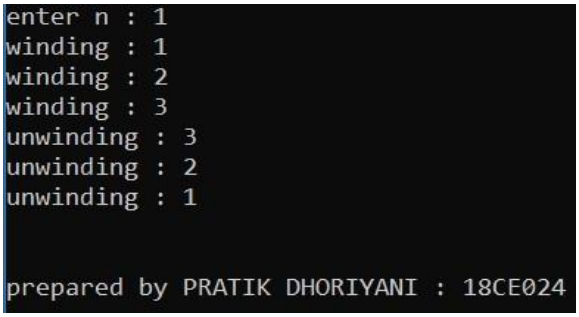
Question: Show stepwise solution of winding and unwinding phase of recursion

CODE:

```
#include<iostream>
void rec(int);
using namespace std;
int main()
{
   int n;
   cout<<"enter n : ";
   cin>>n;
   rec(n);
```

```
   cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n";
   return 0;
}
void rec( int n)
{
   cout<<"winding : "<<n<<"\n";
   if(n<3)
   {
      rec(n+1);
   }
   cout<<"unwinding : "<<n<<"\n";
}
```

OUTPUT:



EXPLANATION:

Recursion is a process of building a loop by calling a fun. Itself.
A terminating condition is set to build limited number of iterations.
The process of forwarding fun. call in recursion is called winding and returning control back on returning value is called unwinding….

5. Write a C++ program to find the number of vowels present in the given character array using pointer arithmetic.

CODE:

```
#include<iostream>
using namespace std;
int main()
{
   char arr[50];
   char *p;
```

```
    int c=0;
    cout<<"enter string : ";
    cin>>arr;
    p=arr;
    while(*p != '\0')
    {
       if(*p=='a' || *p=='e' || *p=='i' || *p=='o' || *p=='u' || *p=='A' || *p=='E' || *p=='I' ||
            *p=='O' || *p=='U')
       {
          c++;
       }
       p++;
    }
    cout<<"the number of vovels in given string is : "<<c;
    cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n";

    return 0;
}
```

OUTPUT:

```
enter string : PRATIK
the number of vovels in given string is :  2

prepared by PRATIK DHORIYANI : 18CE024
```

6. Find error in the following code and give reasons for each error:
Can we declare an array of references? Can we assign NULL value to reference variable? Is **Reference variable** a pointer variable? Can we declare a reference variable without initializing it? Does Reference Variable change the original value of variable?

```
#include<iostream>
using namespace std;
int main()
 {
int no1=10, no2=12;
int & x=no1;
int & r;
int & c = NULL;
int & d[2] = {no1,no2};
cout<<"x = "<< x+20;
cout<<"no1="<< no1+10;
```

```
return 0;
}
```

Solution:

1. **Can we declare an array of references?**
   No , We can not declare array of references…….
   When allocating array of size x,program creates a collection of default initialized objects. Since reference does not have a default value, creating such array is semantically illegal.
2. **Can we assign NULL value to reference variable?**
   No , We can not assign NULL value to reference variable.
3. **Is Reference variable a pointer variable?**
   No , Reference variable is not a pointer variable.
   Because reference variable does not contain any type of extra memory in pc in other side pointer variable contain some particular extra memory in pc.
4. **Can we declare a reference variable without initializing it?**
   No , we can not contain reference variable without initializing it……
   Because it always contains some value….
5. **Does Reference Variable change the original value of variable?**
   Yes , Reference Variable change the original value of variable…..
   Because actually reference variable is same as original variable ….

OUTPUT:

```
#include<iostream>
using namespace std;
int main()
  {
int no1=10, no2=12;
int & x=no1;
int & r;          //without initializing we can not declare reference
int & c = NULL;     //we can not assign null value to reference
int & d[2] = {no1,no2}; //we can not declare array of reference
cout<<"x = "<< x+20;
cout<<"no1="<< no1+10;
return 0;
  }
```

7. Find output of the following code: Explain how **scope Resolution operator** is used to access global version of a variable.

```
#include<iostream>
using namespace std;
int m=30;
int main()
{
int m=20;
{
int m=10;
cout<<"we are in inner block"<<endl;
cout<<"value of m="<<m<<"\n";
cout<<"value of ::m="<<::m<<"\n";
}
cout<<"we are in outer block"<<endl;
cout<<"value of m="<<m<<"\n";
cout<<"value of ::m="<<::m<<"\n";
return 0;
}
```

OUTPUT:

```
we are in inner block
value of m=10
value of ::m=30
we are in outer block
value of m=20
value of ::m=30


prepared by PRATIK DHORIYANI : 18CE024
```

EXPLANATION:

C++ Provides the unary scope resolution (::) to access a globle variable when a local variable of the same name is in scope.

The unary scop resolution operator can not be used to access a local variable of same name in outer block.

8. Find Error in the following code of a program and give explanation why these errors exist. Refer this link for more understanding: http://www.thegeekstuff.com/2012/06/c-constant-pointers/

Note: There is mistake in Balagury 6th edition for syntax of constant pointer and pointer to constant.

1. //This is an example of **constant pointer**	**2.** //This is an example of **pointer to constant**	**3.** //This is an example of **constant pointer to a constant**

#include <iostream> using namespace std;
int main()
{
int var1 = 35,var2 = 20;
int *const ptr = &var1;
ptr = &var2;
cout<<"var1= "<<*ptr;
return 0;
}
	#include <iostream> using namespace std;
	int main()
	{
	int var1 = 43;
	const int* ptr = &var1;
	*ptr = 1;
	var1=34;
	cout<<"var1 = "<< *ptr;
	return 0;
	}
		#include <iostream> using namespace std;
		int main()
		{
		int var1 = 0,var2 = 0;
		const int* const ptr = &var1;
		*ptr = 1;
		ptr = &var2;
		cout<<"Var1 = "<<*ptr; return 0;
		}

EXPLANATION:

1. **Constant Pointer:**
   In this type of pointer we can not change the value of pointer because it's a constant pointer…so here in actual vale of pointer is address of variable…so we cant assign new address to constant pointer…
2. **Pointer to Constant:**
   In this type of pointer pointer is a normal but our variable is a constant …
   so when ever we try to change the value of variable using pointer its gives the error…
   so we cant change the value of variable but we can change the address of pointer…
3. **Constant pointer to a constant:**
   this type of pointer is a combination of 1st and 2nd type of pointer….
   Here we cant change the address of pointer or value of pointer….

9. Write a program to enter a size of array. Create an array of size given by user **using** "**new" Dynamic memory management operator (free store operator).** Enter the data to store in array and display the data after adding 2 to each element in the array. Delete the array by **using "delete" memory management operator.**

CODE:

```
#include<iostream>
using namespace std;
int main()
{
```

```
    int n;
    cout<<"enter size of array : ";
    cin>>n;
    int *p=new int[n];
    for(int i=0;i<n;i++)
    {
       cout<<"enter "<<i<<" element : ";
       cin>>*(p+i);
    }
    cout<<"before modification : \n";
    for(int i=0;i<n;i++)
    {
       cout<<*(p+i)<<"\n";
    }
    cout<<"after modification +2 : \n";
    for(int i=0;i<n;i++)
    {
       cout<<*(p+i)+2<<"\n";
    }
    cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
    return 0;

}
```

OUTPUT:

```
enter size of array : 4
enter 0 element : 1
enter 1 element : 2
enter 2 element : 3
enter 3 element : 4
before modification :
1
2
3
4
after modification +2 :
3
4
5
6

prepared by PRATIK DHORIYANI : 18CE024
```

10. Find the output of following program. Explain the **use of bool data type**.

```
#include<iostream>
using namespace std;
int main()
{
    bool a = 321, b;
    cout << "Bool a Contains : " << a<<endl;
    int c = true;
    int d = false;
    cout<<"c = "<<c <<endl<<"d = "<<d;
    c = a + a;
    cout << "\nInteger c contain : " << c;
    b = c + a;
    cout << "\nBool b contain : " <<b;
    return 0;
}
```

EXPLANATION:

In c++ bool datatype has been introduced to hold a boolean values ,true or false.
The vales true and false have been added as keyword in the c++ .
Default numeric values of true is 1 & false is 0…..

OUTPUT:

```
Bool a Contains : 1
c = 1
d = 0
Integer c contain : 2
Bool b contain : 1
```