**UNITY UNIVERSITY, ADAMA CAMPUS**

**DEPARTEMENT OF COMPUTER SCIENCE**

**FUNDAMENTALS OF PROGRAMMING SELIF-TEST EXERCISE**

/\*Algorithm1. Execution starts from main2. g is initialized to 153. Call function and pass 15 to a function4. When the control move to main, a =155. Then add the value of g and a (g=30)    g=(15+15)6. output the value of g=307. The control back to the main program    and the value of g incremented by 18. So the value of g=31 and output g \*/#include<iostream>using namespace std;//Define function int g=0;void func(){    int a=15;    g=(g+a);    cout<<g<<endl;}//main functionint main (){    g=15;    func();    g++;    cout<<g<<endl;    return 0;}//End of main ()

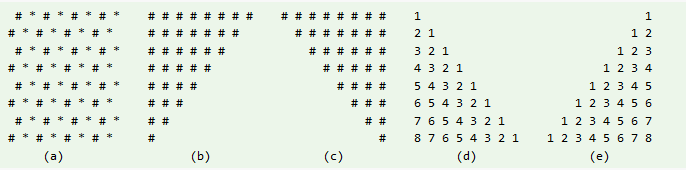
2. Read the following C++ program carefully and determine the output of the program

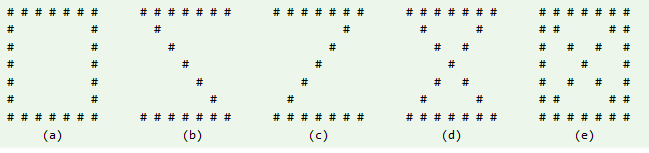
#include<iostream>using namespace std;//Function prototypeint myFunc(int);//Define global variables and initialized with some valueint x=20, y=25;//Main functionint main(){cout<<"\*\*\*\*\*BEGIN THE PROGRAM\*\*\*\*\*\*"<<endl<<endl;cout<<"x globally="<<x<<endl;cout<<"y globally="<<y<<endl<<endl;//Declare variables local to main but this//x and y hide the global variables not changeint x, y;x=15;y=10;cout<<"x local to main="<<x<<endl;cout<<"y local to main="<<y<<endl<<endl;//Call a functionmyFunc(26);cout<<endl<<endl;cout<<" x after back from func call="<<x<<endl;cout<<" y after back from func call="<<y<<endl<<endl;cout<<"\*\*\*\*\*\*\*END OF THE PROGRAM\*\*\*\*\*\*\*\*"<<endl;return 0;}//End of main ()//Define global variable visible to myFunc()int z=30;//Define a function named myFunc()int myFunc(int a){    cout<<"z globally="<<z<<endl;    cout<<"a local to myFunc="<<a<<endl;//Access x and y globallycout<<"x globally accessed="<<x<<endl;cout<<"y globally accessed="<<y<<endl;//Declare variable z that hides z in globalint z=60;cout<<"z local to myFunc="<<z<<endl;//Declare variables within its scope{    int m=70, n=80;    cout<<"m within this block="<<m<<endl;    cout<<"n within this block="<<n<<endl<<endl;//Scopes are nested{    int m=90, n=100;    cout<<"m within nested scope="<<m<<endl;    cout<<"n within nested scope="<<n<<endl<<endl;}//End of nested block}//End of outer block}//End of myFunc()

3. Refer Q2. Write the main program and the function definition in Dev-C++ or RedPanda editor. Run the program and cross check the output of this program with output you observed in Q2.

4. Write the difference between local and global variables shortly by considering the lifetime and when memory is reserved for global and local variables respectively

5. Print these patterns using nested loop (in a program called PrintPattern1x). Use a variable called size for the size of the pattern and try out various sizes. You should use as few printing statements as possible. Write separate C++ program to print the following pattern Hints:  
The equations for major and opposite diagonals are row = col and row + col = size + 1. Decide on what to print above and below the diagonal.



6. Write separate C++ program to print the following pattern