```
//pi.cpp
#include <iostream>
#include <cstdlib>
#include <cmath>
using namespace std;

int main() {
        int terms;

        do {
                cout << "Enter the value of the parameter 'n' in
the Leibniz formula (or -1 to quit):" << endl;
                cin >> terms;

                if(terms == -1) {
                        exit(1);
                }

                int n = 1;
                double pi = 0;
                string termout;

                for(int i = 0;i <= terms;i++) {
                        pi = pi + pow(-1.0,
i)/static_cast<double>(n);
                        n = n + 2;
                }

                pi = 4*pi;

                cout.setf(ios::fixed);
                cout.setf(ios::showpoint);
                cout.precision(3);
                cout << "The approximate value of pi using " <<
terms+1;

                if(terms == 0) {
                        cout << " term is: " << pi << endl;
                }
                else {
                        cout << " terms is: " << pi << endl;
                }
        }
        while(true);
}
```

```
//min4.cpp
#include <iostream>
#include <cstdlib>
using namespace std;

int smallest_of_two(int num1, int num2);
int smallest_of_four(int num1, int num2, int num3, int num4);

int main(int argc, char *argv[]) {
        if (argc != 5) {
    cerr << "Usage: " << argv[0] << " num1 num2 num3 num4" <<
endl;
    cerr << " Prints smallest of the four numbers" << endl;
    exit(1);
  }

        int a = atoi(argv[1]);
  int b = atoi(argv[2]);
  int c = atoi(argv[3]);
  int d = atoi(argv[4]);

  cout << smallest_of_four(a,b,c,d) << endl;

  return 0;
}

int smallest_of_two(int num1, int num2) {
        if (num1 < num2)
    return num1;
  return num2;
}

int smallest_of_four(int num1, int num2, int num3, int num4) {
        return smallest_of_two(smallest_of_two(num1, num2),
smallest_of_two(num3, num4));
}
```

//Computer Functions
Allocates the computer's resources to the different tasks that the computer must accomplish – Operating System
Converts a program from a high-level language to machine language – Compiler
Stores a program while it is being executed – Main memory
Executes a program stored in main memory – Processor
Converts a program from one high-level language to another high-level language – None

```
// starT.cpp   A demonstration of ASCII Art printing T characters

#include <iostream>
#include <cstdlib>
#include <string>
using namespace std;

void assertEquals(string expected, string actual, string message); //removed
string starT(int width, int height);
void runTests(void); //removed

string starT(int width, int height)
{
  string result="";

  if((height < 2) || (width < 3) || (width % 2 == 0)) {
    //Return an empty string
    return result;
  }

  //Line 1 (all stars)
  for(int col = 1; col <= width; col++) {
    result += "*";
  }

  //Add new line
  result += "\n";

  //All other lines
  int spaces;
  spaces = (width-1)/2;
  for(int row = 2; row <= height; row++) {
    //Add beginning spaces
    for(int i = 1; i <= spaces; i++) {
      result += " ";
    }

    //Add star
    result += "*";

    //Add ending spaces
    for(int i = 1; i <= spaces; i++) {
      result += " ";
    }

    //Add new line
    result += "\n";
  }

  return result;
}

int main(int argc, char *argv[]) {
  //Parameter check
  if(argc != 3) {
    cerr << "Usage: " << argv[0] << " width height" << endl;
    exit(1);
  }

  //Get width and height from command line arguments
  int width = stoi(argv[1]);
  int height = stoi(argv[2]);

  //Test code
  if(width == -1 && height == -1) {
    runTests();
    exit(0);
  }

  //Call program, print result on cout
  cout << starT(width, height);
  return 0;
}
```