实验一　C++ 入门

1. 编写程序从键盘输入一个整数和一个实数，输出两者相加的结果。

2. 编写一个程序从键盘输入圆的半径，计算并输出该半径对应圆的面积.

3. 从键盘输入两个字符串给 string 类型的变量，输出这两个字符串的长度和两个字符串拼接后的新字符串。

4. 模仿下面的“输出整数的二进制形式”的程序，输出一个 long 类型数的二进制形式

```
#include<iostream>
#include <iomanip>
#include <bitset>
 using namespacestd;
 intmain(){
 unsigned char x =’011’;
 cout<<" binary format of "<<x<<" is "<<bitset<sizeof(unsigned char)*8>(x)<<endl
 return 0;
 }
```

实验二控制流 Flow of Control

1. 用 switch 语句编写一个相对完整的命令主控程序，用单个字符表示一个命令，根据用户输入命令字符的不同，输出代表不同命令的一个字符串。

```
char cmd;
  switch(cmd){
  case    x’: cout<<”the program will exit!\n”;
              break;
  case ‘p’: cout<<”the program will print some information!”);
              break;
  default: cout<<”the program will do default operation!”);
              break;
  }
```

2. 编写一个程序，从键盘输入一个代表年份的整数（如 1980），输入一句该年份是否润年的一句话。程序框架：

```
#include <iostream>
int main(){
    int year;
    bool is_leap_year;
    std::cin>>year; //等待用户从键盘输入一个代表年份的整数

    //...add your code here...
```

```
    if(is_leap_year)
        std::cout<<year<<" is a leap year!\n";
    else
    std::cout<<year<<" is not a leap year!\n";
}
```

3. 编写一个程序，从键盘输入一个整数 n，该程序计算并输出其阶乘 n! = 1*2*3*…*n 。
4. 编写一个程序，从键盘输入 3 个浮点数，按照从小到大次序输出这三个数.
5. 编写程序，输出如下的图形

1）

```
 1   2   3   4   5   6   7   8   9
 2   4   6   8  10  12  14  16  18
 3   6   9  12  15  18  21  24  27
 4   8  12  16  20  24  28  32  36
 5  10  15  20  25  30  35  40  45
 6  12  18  24  30  36  42  48  54
 7  14  21  28  35  42  49  56  63
 8  16  24  32  40  48  56  64  72
 9  18  27  36  45  54  63  72  81
```

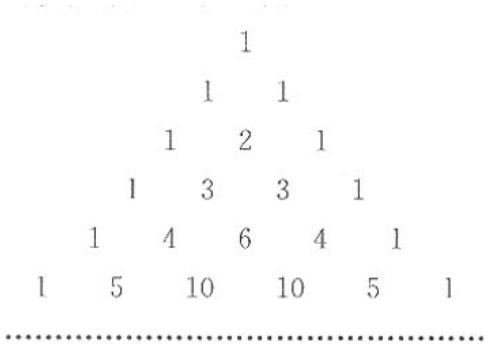
2）

```
                1
              1 2 1
            1 2 3 2 1
          1 2 3 4 3 2 1
        1 2 3 4 5 4 3 2 1
      1 2 3 4 5 6 5 4 3 2 1
    1 2 3 4 5 6 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
```

3)

```
                1
              1 2 1
            1 2 3 2 1
          1 2 3 4 3 2 1
        1 2 3 4 5 4 3 2 1
      1 2 3 4 5 6 5 4 3 2 1
    1 2 3 4 5 6 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1
  1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
    1 2 3 4 5 6 7 6 5 4 3 2 1
      1 2 3 4 5 6 5 4 3 2 1
        1 2 3 4 5 4 3 2 1
          1 2 3 4 3 2 1
            1 2 3 2 1
              1 2 1
                1
```

6. 编写一个程序，从键盘输入一个整数 n，该程序输出杨辉三角的 n 行。(选做)



7. 编写一个如下的“猜数字”游戏。其中红色矩形中的是游戏者输入的猜想字符。程序提示游戏猜想一个介于 1 到 100 之间的整数，游戏者没猜一次的数字如果不正确，就给出提示 "Too high" 或"Too lower"。程序设定了一个最大猜想次数（比如 7 次）。

```
Hi! I'm thinking of a random number between 1 and 100.
--- Attempt 1
Guess what number I am thinking of: 50
Too high.
--- Attempt 2
Guess what number I am thinking of: 25
Too high.
--- Attempt 3
Guess what number I am thinking of: 17
Too high.
--- Attempt 4
Guess what number I am thinking of: 9
Too low.
--- Attempt 5
Guess what number I am thinking of: 14
Too high.
--- Attempt 6
Guess what number I am thinking of: 12
Too high.
--- Attempt 7
Guess what number I am thinking of: 10
Too low.
Aw, you ran out of tries. The number was 11.
```

注：要生存随机数，可以用 rand 这样的库函数（ http://www.cplusplus.com/reference/cstdlib/rand/ ）。

如：rand() % 7

其中 rand()生成一个介于 0 和很大数之间的随机数，而%7 后将得到一个介于 0 和 6 之间的整数。当然你需要包含相应的头文件：#include <stdlib.h>或#include <cstdlib>

实验三函数 functions

1. 编写一个函数计算一个已知半径的圆的面积。函数参数采用引用参数和返回值两种方式返回结果.
2. 理解并完成汉诺塔问题的程序，输出为解决汉诺塔问题，盘子的移动步骤！关于汉诺塔

问题，请查看数据结构的《栈与队列》课件。

```
void move(char x , int    n,    char z){
    cout<<"move the "<<n<<"disk from "<<c<<"to"<<z<<"\n";
}
void hanoi ( int n, char x, char y, char z){
    if (n==1)     move(x, 1, z);
    else {
          hanoi (n-1, x, z, y);
          move(x, n, z);
          hanoi(n-1, y, x, z);
     }
}
int main(){
       //...
       return 0;
}
```

实验四数组 array 和字符串 string

1. 用下标方式实现一些常见的 C 风格字符串的处理函数。如 strcpy,strcat,strlen,strcmp,…. 具体可参见 msdn 对这些函数的说明：http://msdn.microsoft.com/en-us/library/kk6xf663.aspx。

   示例：

```
intStrlen(const char *str){
      inti = 0 ;
      while(str[i]!='\0') i++;
      return i;
 }
```

2. 请设计一个程序，将两个整数数组合并为一个整数数组，要求合并后的数组是从小到大有序排列的。

实验五指针 pointer

1. 阅读并编译下面的程序，改正其中的程序错误

```
void main() {
    int*      x;
    int*      y;

    x = new int;
    *x = 42;

    *y = 13;
```

```
        y = x;
        *y = 13;
    }
```

2. 函数，用于交换两个变量的值，要求用传递变量指针（传值方式）和传递变量引用方式（传引用）写出两个这函数，并编写主函数测试这两个函数。

3. 用指针方式实现一些常见的 C 风格字符串的处理函数。如 strcpy,strcat,strlen,strcmp,…. 具体可参见 msdn 对这些函数的说明：http://msdn.microsoft.com/en-us/library/kk6xf663.aspx。
示例（也可参考 http://www.math.bas.bg/~nkirov/2005/oop/deitel/cpphtp4_05.pdf ）：

```
intStrlen(const char *str){
    char*p = str;;
    while( *p!='\0') p++;
    return p-str;
 }
```

实验六类 class

1 实现一个表示三维向量类 Vector3，能够完成常见的数学向量运算，如加减、点积、差积、求长等。

```
structVector3{
doublex,y,z;
Vector3(double x   = 0, double y = 0, double z = 0) {
    this->x = x;
     //…
  }
double length() {
    //…
  }
};
Vector3 add(constVector3 &u, constVector3 &v);
Vector3 sub(constVector3 &u, constVector3 &v);
double dot(constVector3 &u, constVector3 &v);
Vector3 cross(constVector3 &u, constVector3 &v);
void normalize(constVector3 &u); //规范化，将向量单位长度化！
```

思考：如何将数据成员变为 private？如数据成员成为 private，则如何实现这个修改的类（是否需要增加成员函数？）？

2. 设计一个 Desk 类用于表示一副 52 张的扑克牌，公共接口包含用于洗牌、发牌、显示牌局中各方所拿的牌、比较牌的大小(如 Q 比 J 大)等成员函数，要模拟洗牌，可以用 rand 这样的库函数。

3（拓展实验）链表的实现。实现链表的常见操作如初始化、插入、删除、查询或修改某个

数据元素、求表长等。

```
struct LNode{
    T data;
    LNode *next;
};
class List{
    LNode *head;
public:
    //....
};
```

实验七 继承 Inheritance 与多态 Polymorphism

1. 实现一个公司员工信息管理程序，员工包括雇员和经理两种人员，当然经理也是雇员。请定义至少下述三种类,并对雇员和经理类定义一个 print 虚函数，用以输出员工信息。每个员工有其雇佣日期、姓名等信息。公司的所有员工用一个（雇员基类的）指针向量表示。

```
class Date{
  int year,month,day;
  public:
     Date(int year=2014,int month=5, int day=6);
     Date(string date);
     void getDate(int& year, int &month, int &day);
     void setDate(int year, int month, int day);
     //获取或设置字符串格式的日期，如2014-05-06
     string getDate();
     void setDate(string date);
   //...
};

class Employee{
protected:
    string name;
    Date date;
 public:
    //...
};
class Manager: public Employee{
private:
   int level;
public:
   //...
};
class Company{
   std::vector< Employee *> employees; //请自己查询vector用法
```

```
public:
   //...
};
```

2. 实现一个CardGame类表示一个普通的52张纸牌，包括四种花色，每种花色有13张牌(A、K、Q、J、10、9、...)，该纸牌类能完成通常的扑克牌的功能（如洗牌、发牌、显示牌、比较大小等)，在此基础上，至少实现2种针对特定纸牌玩法的派生类纸牌游戏，如Point24(24点游戏)、five_ten_K(五十K游戏)、Hearts(红心大战)、PushyPorkers（拱猪)、Guandan（惯蛋)。 如五十K游戏规则或玩法：
http://baike.baidu.com/view/558451.htm http://www.appchina.com/app/com.jday.ftk/

实验八 内存管理 Memory

1. （1）实现一个可以存储一系列整数的动态数组类 IntegerArray 或字符串类 String。
（2）用主函数分别创建该类的 Stack 或 heap 存储上的实例并保证在程序结束前或变量使用结束后能干净地释放内存。

2. 实现一个矩阵类 Matrix，要求能实现常用的矩阵运算如拷贝、加减乘、转置、查询或修改某个矩阵元数的值。

```
class Matrix{
   double *data; //用动态分配的数组存储矩阵中的元素
   int rows,cols;
public:
   Matrix(int  rows=0, int cols = 0);
   Matrix(const Matrix &m);//拷贝构造函数
   double get( int row, int col);
   void set( int row, int col, double value);
   Matrix transpose(); //转置
   friend Matrix add(const Matrix A, const Matrix& B);
   //....
}
```

实验九 运算符重载 Operator Overloading

1. 模仿标准库的字符串 string，实现一个自己的 String 类，要求重载运算符+，=，==，[]等。

```
class String{
   //...
   friend ostream& operator<<(ostream &O,String S);

};
//重载输出流运算符 operator<<
ostream& operator<<(ostream &O,String S){
      //...
     return O;
```

```
}

    int main(){
        String S("Hello "), T = "world";
        String W = S+T;
        if (W=="hello world")
            cout<<"W is equal to: hello world"<<"\n";
        cout<<"the length of W is: "<<W.size() <<endl;
        cout<<W<<"\n";
        //....
         return 0;
    }
```

2. 实现一个重载赋值和函数运算符的矩阵类 Matrix，要求重载函数运算符根据下标访问每个数据元素.如

```
    class Matrix{
        //....
    };
    int main(){
        Matrix M(2,3),N(3,4);
        M(0,2) = 20;
        N = M;
        //....
         return 0;
    }
```

实验十 模板 Template

1. 模仿标准库的 vector 模板，实现一个可以存储一系列同类型数据元素的向量类 Vector. 实现常用的一些操作，如下标运算符[], push_back(), size(), insert(),erase()等。

```
tempalte<typename T>
class Vector{
        T *data;
        int length;
        int capacity;
 public:
    //...
};
int main(){
        Vector<int> ints;
        Vector<double> doubles;
        doubles.push_back(40.5)
        for(int i = 0 ; i<ints.size();i++)    cout<< ints[i]<<" ";
        return 0;
```

```
}
```