1．实验目的

（1）掌握在设计条件型循环结构时如何正确地设定循环条件，以及如何控制循环的次数。

（2）了解条件型循环结构的基本测试方法。

（3）掌握如何正确地控制计数型循环结构的次数。

（4）了解对计数型循环结构进行测试的基本方法。

（5）了解在嵌套循环结构中提高程序运行效率的方法。

2．实验内容

（1）输入一个正整数，并将其颠倒过来，例如12345颠倒后为54321.

（2）将一个长整型数s的每一位数位上的偶数依次取出来，构成一个新的数t，其高位仍在高位，低位仍在低位。例如，s=87653142时，t中的数为8642.

（3）判断101～200范围内有多少个素数。

（4）编写程序，输入杨辉三角形。

3．实验步骤和实验结果

1. 问代码如下：

|  |
| --- |
| #include<stdio.h>  **int** main()  {  **int** n,t=0;  printf("输入一个正整数：");  **do**  {  scanf("%d",&n);  **if**(n<0)  printf("输入数据不正确，请重新输入！\n");  }**while**(n<0);  **while**(n)  {  t=t\*10+n%10;  n=n/10;  }  printf("颠倒过来的数为：%d",t);  **return** 0;  } |

在Xcode中输入上面代码结果如下：



要使负数也能得到颠倒后的数，对上面代码进行适当修改，修改后的程序如下：

|  |
| --- |
| #include<stdio.h>  int main()  {  int n,t=0;  printf("输入一个整数：");  scanf("%d",&n);  if(n==0)  t=0;  else if (n>0)  {  while(n)  {  t=t\*10+n%10;  n=n/10;  }  }  else  {  n=-n;  while(n)  {  t=t\*10+n%10;  n=n/10;  }  t=-t;  }    printf("颠倒过来的数为：%d",t);  return 0;  } |

运行结果如下：



（2）问代码如下：

|  |
| --- |
| #include<stdio.h>  int main()  {  long int s,res=1,t=0;  printf("输入一个整数：");  scanf("%ld",&s);  while(s)  {  if((s%10)%2==0)  {  t=t+(s%10)\*res;  res=res\*10;  }  s/=10;  }  printf("得到的数为: %ld\n",t);  return 0;  } |

在Xcode中输入上面代码结果如下：



（3）问代码如下：

|  |
| --- |
| #include<stdio.h>  #include <math.h>  int main()  {  int i,j,k,count=0;  for(i=101;i<=200;i++)  {  k=sqrt(i);  for(j=2;j<=k;j++)  if(i%j==0)  break;  if(j>k)  count++;  }  printf("%d\n",count);  return 0;  } |

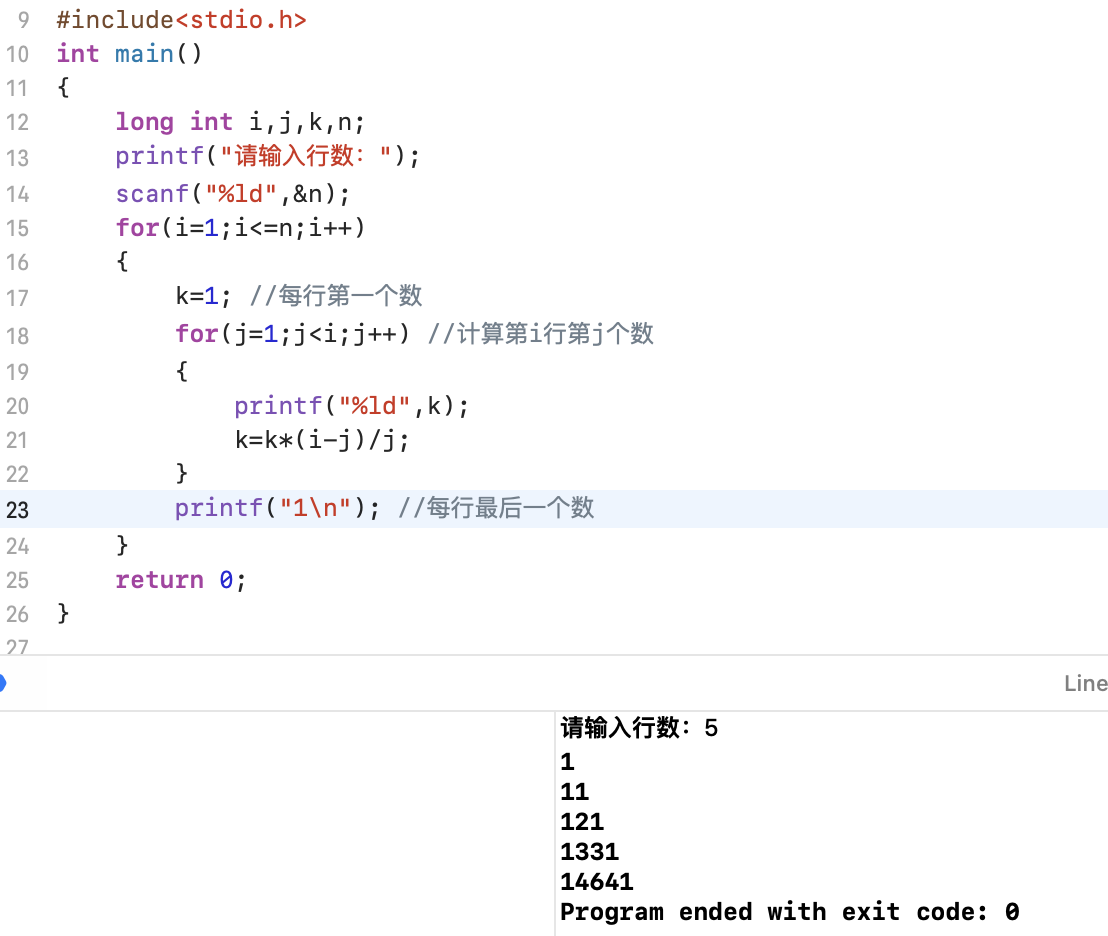
在Xcode中输入上面代码结果如下：



（4）问代码如下：

|  |
| --- |
| #include<stdio.h>  int main()  {  long int i,j,k,n;  printf("请输入行数：");  scanf("%ld",&n);  for(i=1;i<=n;i++)  {  k=1; //每行第一个数  for(j=1;j<i;j++) //计算第i行第j个数  {  printf("%ld",k);  k=k\*(i-j)/j;  }  printf("1\n"); //每行最后一个数  }  return 0;  } |

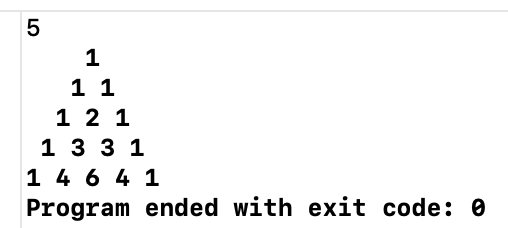
在Xcode中输入上面代码结果如下：



改为等腰三角形输入（我只会用数组的形式）

|  |
| --- |
| #include<stdio.h>  #define N 1000  int main( )  {  int a[N][N]; //定义足够大的数组  int i,j;  int n;  scanf("%d",&n);  a[1][1]=1; //给杨辉三角赋予初值  a[2][1]=1;  a[2][2]=1;  for(i=3;i<=99;i++) //先算出杨辉三角值  {  for(j=1;j<=i;j++)  a[i][j]=a[i-1][j-1]+a[i-1][j];  }  for(i=1;i<=n;i++)  {  int k=0;  for(j=i;j<=n-1;j++)  printf(" "); //因为要为等腰三角形形状，故打印前面的空格  for(j=1;j<=2\*i-1;j++)  {  if(j%2)//奇数位打印  printf("%d",a[i][++k]);//打印具体值  else  printf(" ");//打印数与数之间的间隙  }  printf("\n");  }  return 0;  } |

结果如下：



4．分析与讨论

（1）总结条件循环结构的一般方法

（2）如何测试计数型循环结构的控制表达式中的错误

（3）从实验中你得到了哪些提高嵌套循环程序效率的启示

答：循坏常用for和while函数，注意，要设置条件能使循环终止。