**青少年软件编程（C语言）等级考试试卷（三级）**

2020年9月 分数：100  题数：5

一、编程题(共5题，共100分)

|  |  |
| --- | --- |
| 1. | 因子问题 |
|  | **因子问题**  任给两个正整数N、M，求一个最小的正整数a，使得a和(M-a)都是N的因子。  时间限制：10000  内存限制：65536  **输入**  包括两个整数N、M。N不超过1,000,000。  **输出**  输出一个整数a，表示结果。如果某个案例中满足条件的正整数不存在，则在对应行输出-1  **样例输入**  35 10  **样例输出**  5 |
|  | #include<iostream>  using namespace std;  int main(){  int n,m;  cin>>n>>m;  for(int i=1;i<m;i++){  if(n%i==0&&n%(m-i)==0){  cout<<i;  return 0;  }  }  cout<<"-1";  return 0;  } |

|  |  |
| --- | --- |
| 2. | 质数的和与积 |
|  | **质数的和与积**  两个质数的和是S，它们的积最大是多少？  时间限制：10000  内存限制：65536  **输入**  一个不大于10000的正整数S，为两个质数的和。  **输出**  一个整数，为两个质数的最大乘积。数据保证有解。  **样例输入**  50  **样例输出**  589 |
|  | #include<iostream>  #include<cstdio>  #include<string>  #include<math.h>  #include<cstring>  using namespace std;  int prime(int a)  {  int i;  if(a==1)  return 0;  for(i=2;i<=sqrt(a);i++)  if(a%i==0)  return 0;  return 1;  }  int main()  {  int j,s,x=0,y;  scanf("%d",&s);  y=s;  for(j=2;j<=s/2;j++)  {  if(prime(j)==1&&prime(s-j)==1)  {  if(j>x&&(s-j)<y)  {  x=j;  y=s-j;  }  }  }  printf("%d",x\*y);  return 0;  } |

|  |  |
| --- | --- |
| 3. | 扩号匹配问题 |
|  | **扩号匹配问题**  在某个字符串（长度不超过100）中有左括号、右括号和大小写字母；规定（与常见的算数式子一样）任何一个左括号都从内到外与在它右边且距离最近的右括号匹配。写一个程序，找到无法匹配的左括号和右括号，输出原来字符串，并在下一行标出不能匹配的括号。不能匹配的左括号用"$"标注,不能匹配的右括号用"?"标注.  时间限制：3000  内存限制：65536  **输入**  输入包括多组数据，每组数据一行，包含一个字符串，只包含左右括号和大小写字母，**字符串长度不超过100**  **输出**  对每组输出数据，输出两行，第一行包含原始输入字符，第二行由"$","?"和空格组成，"$"和"?"表示与之对应的左括号和右括号不能匹配。  **样例输入**  ((ABCD(x)  )(rttyy())sss)(  **样例输出**  ((ABCD(x)  $$  )(rttyy())sss)(  ? ?$ |
|  | #include<stdio.h>  #include<string.h>  int main(int argc, char \*argv[])  {  char str[110];  int len,i,k;  int stack[110],end=0;  while(scanf("%s",str)!=EOF)  {  printf("%s\n",str);  end=0;  len=strlen(str);  for(i=0;i<len;i++)  {  if(str[i]!='('&&str[i]!=')') str[i]=' ';  }  for(i=0;i<len;i++)  {  if(str[i]=='(')  {  stack[end]=i;  end++;  }  else if(str[i]==')')  {  if(end>0)//栈不为空，则  {  k=stack[end-1];  str[k]=' ';//栈顶的左括号和新遇到的右括号匹配  str[i]=' ';  end--;//栈顶元素出栈  }  else  {  str[i]='?';//右括号不匹配  }  }  }  while(end>0)  {  k=stack[end-1];  str[k]='$';  end--;  }  printf("%s\n",str);  }  return 0;  } |

|  |  |
| --- | --- |
| 4. | 吃糖果2 |
|  | **吃糖果2**  现有n(50 > n > 0)个糖果,每天只能吃2个或者3个，请计算共有多少种不同的吃法吃完糖果。  时间限制：1000  内存限制：65536  **输入**  输入的每一行包括一组测试数据，即为糖果数n。最后一行为0，表示测试结束。  **输出**  每一行输出对应一行输入的结果，即为吃法的数目。  **样例输入**  1  2  3  4  12  0  **样例输出**  0  1  1  1  12 |
|  | #include<iostream>  using namespace std;  int work(int x)  {  if(x==2||x==3){  return 1;  }  if(x==1){  return 0;  }  return work(x-2)+work(x-3);  }  int main(){  int a=1;  while(a==1){  int n;  cin>>n;  if(n==0){  return 0;  }  cout<<work(n)<<endl;  }  return 0;  } |

|  |  |
| --- | --- |
| 5. | 铺砖 |
|  | **铺砖**  对于一个2行N列的走道。现在用1\*2,2\*2的砖去铺满。问有多少种不同的方式。  时间限制：3000  内存限制：131072  **输入**  整个测试有多组数据，请做到文件底结束。每行给出一个数字N，0 <= n <= 250  **输出**  如题  **样例输入**  2  8  12  100  200  **样例输出**  3  171  2731  845100400152152934331135470251  1071292029505993517027974728227441735014801995855195223534251 |
|  |  |