|  |  |  |
| --- | --- | --- |
| 题 | 题目 | 程序 |
| 01 | 对数组中的数据，正向输入后，反向输出 | void main( )  {  int i,a[10];  for(i=0;i<10;i++)scanf(“%d”,&a[i] );  for(i=9;i>=0;i--)printf(“%5d”,a[i]);  printf(“\n”);  } |
| 02 | 三个数中找出一个最大的 | void main()  {  int a,b,c,max;  scanf(“%d%d%d”,&a,&b,&c);  printf(“a=%d,b=%d,c=%d ”,a,b,c);  if(a>b)max=a;  else max=b;  if(c>max)max=c;  printf(“max=%d\n”,max);  } |
| 03 | 找出水仙花数 | void main( )  {  int n,a,b,c;  for(n=100;n<=999;n++)  {  a=n%10;  b=n/10%10;  c=n/100;  if(a\*a\*a+b\*b\*b+c\*c\*c==n)  printf("%5d",n);  }  } |
| 04 | 求最大公约和最小公倍 | 方法之一：在主函数中，最大公约使用循环  void main()  {  int m=24,n=36,mn;  mn=m\*n;  while(m!=n)  if(m>n)m-=n;  else n-=m;  printf(“gcd=%d,gcp=%d\n”,m,mn/m);  }  方法之二：主函数分别调用两个函数求最大公约（递归方法）和最小公倍  int gcd(int m , int n) /\* 求最大公约数 \*/  {  if(m==n) return m;  else if(m>n)return gcd(m-n,n);  else return(m,n-m);  }  int gcp(int m , int n) /\* 求最小公倍数 \*/  {  return m\*n/gcd(m,n);  }  void main( )  {  int a,b;  sacnf(“%d%d”,&a,&b);  printf(“a=%d,b=%d,最大公约数=%d,  最小公倍数=%d\n”,a,b,gcd(a,b),gcp(a,b));  } |
| 05 | 对成绩五级评分 | 之一：分数为整型  void main()  {  int score;  scanf(“%d”,&score);  printf(“score=%d “,score);  switch(score/10)  {  case 10:  case 9:printf(“优秀\n”);break;  case 8: printf(“良好\n”);break;  case 7: printf(“中等\n”);break;  case 6: printf(“及格\n”);break;  default: printf(“不及格\n”);  }  }  输入数据：xx↙  之二：分数为实型  void main()  {  float score;  scanf(“%f”,&score);  printf(“score=%.1f “,score);  switch((int)(score/10))  { case 10:  case 9:printf(“A\n”);break;  case 8:printf(“B\n”);break;  case 7:printf(“C\n”);break;  case 6:printf(“D\n”);break;  default:printf(“E\n”);  }  }输入数据：xx.x↙ |
| 06 | 输入一串字符（换行作为结束）统计元音（a,A;e,E;i,I;o,O;u,U）各自出现的次数 | #include<stdio.h>  void main( )  {  char ch;  int a=0,e=0,i=0,o=0,u=0;  printf(“输入一串字符，换行作为结束：”);  while((ch=getchar( ))!='\n')  switch(ch)  {  case 'a':  case 'A': a++;break;  case 'e':  case 'E': e++;break;  case 'i':  case 'I': i++;break;  case 'o':  case 'O': o++;break;  case 'u':  case 'U': u++;  }  printf("a,A=%d\ne,E=%d\ni,I=%d\no,O=%d\nu,U=%d\n",a,e,i,o,u);  } |
| 07 | 正序和反序打印大小写英文字母 | void main( )  {  char ch;  for(ch=’A’;ch<=’Z’;ch++)  printf(“%c”,ch);  printf(“\n”);  for(ch=’z’;ch>=’a’;ch--)  printf(“%c”,ch);  printf(“\n”);  } |
| 08-1 | 选择法排序 | #define N 10  void main( )  {  int i,j,t,a[N]={1,0,2,9,8,4,7,6,5,3};  for(i=0;i<N-1;i++)  for(j=i+1;j<N;j++)  if(a[i]>a[j]){t=a[i];a[i]=a[j];a[j]=t;}  for(i=0;i<N;i++)printf("%5d",a[i]);  printf("\n");  } |
| 08-2 | 冒泡法排序 | #define N 10  void main( )  {  int i,j,t,a[N];  for(i=0;i<N;i++)scanf("%d",&a[i]);  for(i=0;i<N-1;i++)  for(j=0;j<N-1-i;j++)if(a[j]>a[j+1])  t=a[j],a[j]=a[j+1],a[j+1]=t;  for(i=0;i<N;i++)printf("%5d",a[i]);  printf("\n");  } |
| 09 | 九九表  1  2 4  3 6 9  4 8 12 16  ……………………….. | . void main()  {  int i,j;  for(i=1;i<=9;i++)  {  for(j=1;j<=i;j++)printf(“%4d”,j\*i);  printf(“\n”);  }  } |
| 10 | 字符串输出 | . void main()  {  char www[ ]=”World\0Wide\0Web”;  printf(“%s\n”,www);  } |
| 11 | 斐波那契数列部分项输出 | void main()  {  int i,f[10]={1,1};  for(i=2;i<10;i++)f[i]=f[i-1]+f[i-2];  for(i=0;i<10;i++)  {  printf(“%5d”,f[i]);  if((i+1)%5==0)printf(“\n”);  }  } |
| 12 | 任意输入三条边(a,b,c)，若能构  成三角形计算其面积，判断是  否为等腰、等边、直角或任意,  否则输出“不能构成三角形” | #include<math.h>  void main( )  {  float a,b,c,t,area;  scanf("%f%f%f",&a,&b,&c);  printf("a=%.2f,b=%.2f,c=%.2f ",a,b,c);  if(a+b>c&&b+c>a&&a+c>b)  {  t=(a+b+c)/2.0; area=sqrt(t\*(t-a)\*(t-b)\*(t-c));  printf("area=%.4f ",area);  if(a==b&&b==c)printf(" 等边\n");  else if(a==b||b==c||a==c)printf(" 等腰\n");  else if(a\*a+b\*b==c\*c||a\*a+c\*c==b\*b||b\*b+c\*c==a\*a)  printf(" 直角\n");  else printf(" 任意\n");  }  else printf(" 不能构成三角形!\n");  } |
| 13 | 对下列分段函数，输入x的值后，计算并输出y的值（精确到小数点后2位）  x 当x<1  y= 2x-1 当1≤x<10  3x-11 当x≥10 | void main()  {float x,y;  scanf(“%f”,&x);  if(x<1)y=x;  else if(x<10)y=2\*x-1;  else y=3\*x-11;  printf(“x=%.2f,y=%.2f\n”,x,y);  } |
| 14 | 二维数组的输出和转置 | void main( )  {  int i,j,a[2][3]={{1,2,3},{4,5,6}},b[3][2];  printf("矩阵A:\n\n");  for(i=0;i<2;i++)  {  for(j=0;j<3;j++)  {  printf("%5d",a[i][j]);  b[j][i]=a[i][j];  }  printf("\n");  }  printf("矩阵B:\n\n");  for(i=0;i<3;i++)  {  for(j=0;j<2;j++)printf("%5d",b[i][j]);  printf("\n");  }  } |
| 15 | 计算  的值（精确到小数点后6位） | #define EPS 1E-6  #include<math.h>  void main( )  {  int sign=1;  float n=1.0,t=1.0,Pi=0;  while( fabs(t)>=EPS)  {  Pi+=t; n+=2;  sign=-sign;  t=sign/n;  }  printf("Pi=%.6f\n",4\*Pi);  } |
| 16 | 指针作函数参数 | void swap1(int x,int y)  {  int t;  t=x;x=y;y=t;  }  void swap2(int \*x,int \*y)  {  int t;  t=\*x,\*x=\*y,\*y=t;  }  #define PR printf(“a=%d,b=%d\n”,a,b);  void main()  {  int a=10,b=20;PR  swap1(a,b);PR  swap2(&a,&b);PR  } |
| 17 | （1）编写一个计算摄氏（C）到华氏（F）转换的函数。函数头是：float temp（float c）  计算公式：F=  （2）编写主函数从键盘输入摄氏温度的值，调用上述函数输出摄氏及华氏温度的值（保留两位小数）。 | float temp(float c)  {  retuen 9.0/5.0\*c+32.0;  }  void main()  {  float C,F;  scanf(“%f”,&C);  F=temp(C);  printf(“C=%.2f, F=%.2f\n”,C,F);  } |
| 18 | 有一个磁盘文件，第一次将它的内容显示在屏幕上，第二次把它复制到另一文件上。 | #include<stdio.h>  void main()  {  FILE \*fp1,\*fp2;  fp1=fopen(“file1.c”,”r”);  fp2=fopen(“file2.c”,”w”);  while(!feof(fp1))putchar(fgetc(fp1));  rewind(fp1);  while(!feof(fp1))fputc(fgetc(fp1),fp2);  fclose(fp1); fclose(fp2);  } |
| 19 | 输入实数x，按下列公式计算并输出x和y的值（保留两位小数） | #include<math.h>  main()  {  float x,y;  scanf(“%f”,&x);  if(x<=0)y=sin(x);  else if(x<10)y=x\*x+1;  else 1.0/(x\*x\*x+x\*x+1);  printf(“x=%.2f,y=%.2f\n”,x,y);  } |
| 20-1 | 一维数组输出杨辉三角形 | #define N 10  void main( )/\* 一维数组 \*/  {  int i,j,yh[N];  for(i=0;i<N;i++)  {  yh[i]=1;  for(j=i-1;j>=1;j--)yh[j]+=yh[j-1];  for(j=1;j<=15-i;j++)printf(" ");  for(j=0;j<=i;j++)printf("%4d",yh[j]);  printf("\n");  }  } |
| 20-2 | 二维数组输出杨辉三角形 | #define N 10  main( )  {  int i,j,yh[N][N];  for(i=0;i<N;i++)  {  yh[i][0]=yh[i][i]=1;  for(j=1;j<i;j++)yh[i][j]=yh[i-1][j-1]+yh[i-1][j];  }  for(i=0;i<N;i++)  {  for(j=1;j<=15-i;j++)printf(" ");  for(j=0;j<=i;j++)printf("%4d",yh[i][j]);  printf("\n");  }  } |
|  | 一、用循环语句按下列形式输出九行的数字三角形  1  1 2 1  2 3 2 1  1 2 3 4 3 2 1  …………………………………………..  1 2 3 4……………8 9 8 7……….3 2 1 |  |
|  | 二、将九九表用循环语句先放入二维数组中，再从数组中按下列形式输出  1  4  6 9  ………………….  9 18 27 ………….81 |  |
|  | 1. 按下列公式用递归算法生成和输出Fibonacci数列前20项，每行输出5个数     1 2 3 5  8 13 21 34 55  …………………… |  |