

3.1 写出下面各逻辑表达式的值。设 $a=3, b=4, c=5$ 。

- (1) $a+b>c \ \&\& \ b==c$
- (2) $a \parallel b+c \ \&\& \ b-c$
- (3) $!(a>b) \ \&\& \ !c \parallel 1$
- (4) $!(x=a) \ \&\& \ (y=b) \ \&\& \ 0$
- (5) $!(a+b)+c-1 \ \&\& \ b+c/2$

解:

- (1) 0
- (2) 1
- (3) 1
- (4) 0
- (5) 1

3.2 有 3 个整数 a, b, c , 由键盘输入, 输出其中最大的数, 请编程序。

解: 方法一。N-S 图见图 3-1。

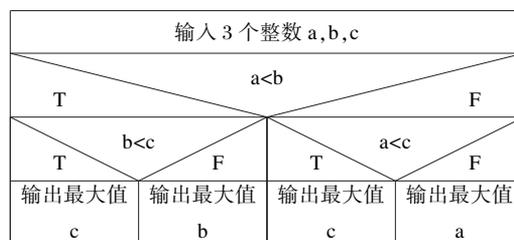


图 3-1

编写程序如下。

```
#include<stdio.h>
int main()
{ int a,b,c;
  printf("请输入 3 个整数:");
```

```

scanf("%d,%d,%d",&a,&b,&c);
if (a<b)
    if (b<c)
        printf("max=%d\n",c);
    else
        printf("max=%d\n",b);
else if (a<c)
    printf("max=%d\n",c);
else
    printf("max=%d\n",a);
return 0;
}

```

运行结果:

请输入3个整数: 12, 34, 9 ↙
max=34

方法二。使用条件表达式,可以使程序更加简明、清晰。

```

#include<stdio.h>
int main()
{ int a,b,c,temp,max;
  printf("请输入3个整数:");
  scanf("%d,%d,%d",&a,&b,&c);
  temp=(a>b)? a:b;           /* 将a和b中的大者存入temp中 */
  max=(temp>c)? temp:c;     /* 将a和b中的大者与c比较,取最大者 */
  printf("3个整数的最大数是%d\n",max);
  return 0;
}

```

运行结果:

请输入3个整数: 12, 34, 9 ↙
3个整数的最大数是34

3.3 有一个函数:

$$y = \begin{cases} x & x < 1 \\ 2x - 1 & 1 \leq x < 10 \\ 3x - 11 & x \geq 10 \end{cases}$$

写程序,输入x,输出y值。

解: 编写程序如下。

```

#include<stdio.h>
int main()
{ int x,y;
  printf("输入x:");
  scanf("%d",&x);

```

```
    if(x<1)                                //x<1
    { y=x;
      printf("x=%3d, y=x=%d\n", x, y);
    }
    else if(x<10)                           //1<=x<10
    { y=2 * x-1;
      printf("x=%d, y=2 * x-1=%d\n", x, y);
    }
    else                                     //x>=10
    { y=3 * x-11;
      printf("x=%d, y=3 * x-11=%d\n", x, y);
    }
    return 0;
}
```

运行结果:

- ① 输入 x: 4 ✓
x=4, y=2 * x-1=7
- ② 输入 x: -1 ✓
x=-1, y=x=-1
- ③ 输入 x: 20 ✓
x= 20, y=3 * x-11=49

3.4 给出一百分制成绩,要求输出成绩等级'A', 'B', 'C', 'D', 'E'。90分及以上为'A', 80~89分为'B', 70~79分为'C', 60~69分为'D', 60分以下为'E'。

解: 编写程序如下。

```
#include<stdio.h>
int main()
{ float score;
  char grade;
  printf("请输入学生成绩:");
  scanf("%f", &score);
  while (score>100 || score<0)
  {printf("\n 输入有误,请重新输入");
   scanf("%f", &score);
  }
  switch((int) (score/10))
  {case 10:
    case 9: grade='A';break;
    case 8: grade='B';break;
    case 7: grade='C';break;
    case 6: grade='D';break;
    case 5:
    case 4:
    case 3:
```

```

    case 2:
    case 1:
    case 0: grade='E';
    }
    printf("成绩是 %5.1f, 相应的等级是%c.\n", score, grade);
    return 0;
}

```

运行结果:

- ① 请输入学生成绩: 90.5 ✓
成绩是 90.5, 相应的等级是 A
- ② 请输入学生成绩: 59 ✓
成绩是 59.0, 相应的等级是 E

 **说明:** 对输入的数据进行检查, 如小于 0 或大于 100, 要求重新输入。(int)(score/10) 的作用是将 (score/10) 的值进行强制类型转换, 得到一个整型值。例如, 当 score 的值为 78 时, (int)(score/10) 的值为 7。然后在 switch 语句中执行 case 7 中的语句, 使 grade='C'。

3.5 给一个不多于 5 位的正整数, 要求:

- ① 求出它是几位数;
- ② 分别输出每一位数字;
- ③ 按逆序输出各位数字, 例如原数为 321, 应输出 123。

解: 编写程序如下。

```

#include<stdio.h>
#include<math.h>
int main()
{ long int num;
  int indiv, ten, hundred, thousand, ten_thousand, place;
                                     /* 分别代表个位、十位、百位、千位、万位和位数 */
  printf("请输入一个整数(0~99999):");
  scanf("%ld", &num);
  if (num>9999)
    place=5;
  else if (num>999)
    place=4;
  else if (num>99)
    place=3;
  else if (num>9)
    place=2;
  else place=1;
  printf("位数:%d\n", place);
  printf("每位数字为:");
  ten_thousand=num/10000;

```

```
thousand=(int)(num-ten_thousand*10000)/1000;
hundred=(int)(num-ten_thousand*10000-thousand*1000)/100;
ten=(int)(num-ten_thousand*10000-thousand*1000-hundred*100)/10;
indiv=(int)(num-ten_thousand*10000-thousand*1000-hundred*100-ten*10);
switch(place)
{case 5:printf("%d,%d,%d,%d,%d",ten_thousand,thousand,hundred,ten,
indiv);
printf("\n 反序数字为:");
printf("%d%d%d%d%d\n",indiv,ten,hundred,thousand,ten_thousand);
break;
case 4:printf("%d,%d,%d,%d",thousand,hundred,ten,indiv);
printf("\n 反序数字为:");
printf("%d%d%d%d\n",indiv,ten,hundred,thousand);
break;
case 3:printf("%d,%d,%d",hundred,ten,indiv);
printf("\n 反序数字为:");
printf("%d%d%d\n",indiv,ten,hundred);
break;
case 2:printf("%d,%d",ten,indiv);
printf("\n 反序数字为:");
printf("%d%d\n",indiv,ten);
break;
case 1:printf("%d",indiv);
printf("\n 反序数字为:");
printf("%d\n",indiv);
break;
}
return 0;
}
```

运行结果:

请输入一个整数(0~99999): 98423 ✓

位数: 5

每位数字为: 9,8,4,2,3

反序数字为: 32489

3.6 企业发放的奖金根据利润提成。利润 I 低于或等于 100 000 元的,奖金可提成 10%;利润高于 100 000 元,低于 200 000 元($100\,000 < I \leq 200\,000$)时,低于 100 000 元的部分按 10% 提成,高于 100 000 元的部分,可提成 7.5%; $200\,000 < I \leq 400\,000$ 时,低于 200 000 元的部分仍按上述办法提成(下同)。高于 200 000 元的部分按 5% 提成; $400\,000 < I \leq 600\,000$ 元时,高于 400 000 元的部分按 3% 提成; $600\,000 < I \leq 1\,000\,000$ 时,高于 600 000 元的部分按 1.5% 提成; $I > 1\,000\,000$ 时,超过 1 000 000 元的部分按 1% 提成。从键盘输入当月利润 I ,求应发奖金总数。要求:

- (1) 用 if 语句编程序;
- (2) 用 switch 语句编程序。

解: 编写程序如下。

- (1) 用 if 语句编程序。

```
#include<stdio.h>
int main()
{ long i;
  double bonus, bon1, bon2, bon4, bon6, bon10;
  bon1=100000 * 0.1;
  bon2=bon1+100000 * 0.075;
  bon4=bon2+100000 * 0.05;
  bon6=bon4+100000 * 0.03;
  bon10=bon6+400000 * 0.015;
  printf("请输入利润 i:");
  scanf("%ld", &i);
  if (i<=100000)
    bonus=i * 0.1;
  else if (i<=200000)
    bonus=bon1+(i-100000) * 0.075;
  else if (i<=400000)
    bonus=bon2+(i-200000) * 0.05;
  else if (i<=600000)
    bonus=bon4+(i-400000) * 0.03;
  else if (i<=1000000)
    bonus=bon6+(i-600000) * 0.015;
  else
    bonus=bon10+(i-1000000) * 0.01;
  printf("奖金是: %10.2f\n", bonus);
  return 0;
}
```

运行结果:

```
请输入利润 i: 234000 ↵
奖金是: 19200.00
```

此题的关键在于正确写出每一区间的奖金计算公式,例如,利润在 10 万~20 万元时,奖金应由两部分组成:

- ① 利润为 10 万元时应得的奖金,即 10 万元 * 0.1。
 - ② 10 万元以上部分应得的奖金,即 $(\text{num}-10 \text{ 万}) * 0.075$ 元。
- 同理,20 万~40 万元这个区间的奖金也应由两部分组成:
- ① 利润为 20 万元时应得的奖金,即 10 万元 * 0.1+10 万元 * 0.075。
 - ② 20 万元以上部分应得的奖金,即 $(\text{num}-20 \text{ 万}) * 0.05$ 元。

程序中先把 10 万元、20 万元、40 万元、60 万元、100 万元各关键点的奖金计算出来, 即 bon1 , bon2 , bon4 , bon6 , bon10 。然后再加上各区间附加部分的奖金即可。

(2) 用 `switch` 语句编写程序, N-S 图见图 3-2。

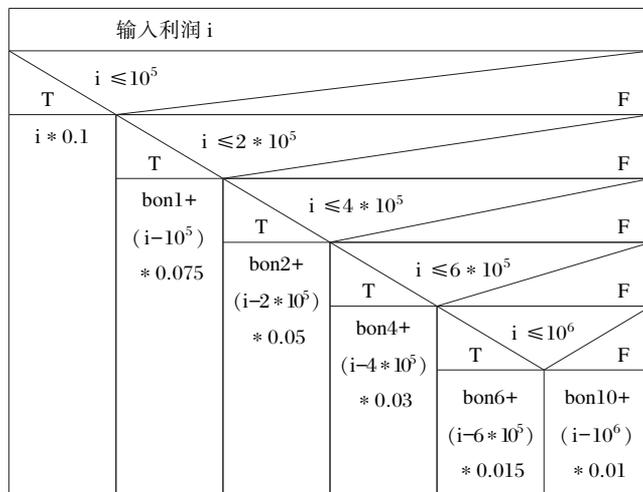


图 3-2

编写程序如下。

```
#include<stdio.h>
int main()
{ long i;
  double bonus,bon1,bon2,bon4,bon6,bon10;
  int branch;
  bon1=100000 * 0.1;
  bon2=bon1+100000 * 0.075;
  bon4=bon2+200000 * 0.05;
  bon6=bon4+200000 * 0.03;
  bon10=bon6+400000 * 0.015;
  printf("请输入利润 i:");
  scanf("%ld",&i);
  branch=i/100000;
  if (branch>10) branch=10;
  switch(branch)
  { case 0:bonus=i * 0.1;break;
    case 1:bonus=bon1+(i-100000) * 0.075;break;
    case 2:
    case 3: bonus=bon2+(i-200000) * 0.05;break;
    case 4:
    case 5: bonus=bon4+(i-400000) * 0.03;break;
    case 6:
    case 7:
    case 8:
```

```
        case 9: bonus=bon6+(i-600000) * 0.015;break;
        case 10: bonus=bon10+(i-1000000) * 0.01;
    }
    printf("奖金是 %10.2f\n",bonus);
    return 0;
}
```

运行结果:

请输入利润 i: 156890 ✓
奖金是: 14266.75

3.7 输入 4 个整数,要求按由小到大的顺序输出。

解: 此题采用依次比较的方法排出其大小顺序。在学习了循环和数组以后,可以掌握更多的排序方法。

编写程序如下。

```
#include<stdio.h>
int main()
{ int t,a,b,c,d;
  printf("请输入 4 个数:");
  scanf("%d,%d,%d,%d",&a,&b,&c,&d);
  printf("a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
  if (a>b)
    { t=a;a=b;b=t;}
  if (a>c)
    { t=a;a=c;c=t;}
  if (a>d)
    { t=a;a=d;d=t;}
  if (b>c)
    { t=b;b=c;c=t;}
  if (b>d)
    { t=b;b=d;d=t;}
  if (c>d)
    { t=c;c=d;d=t;}
  printf("排序结果如下: \n");
  printf("%d %d %d %d \n",a,b,c,d);
  return 0;
}
```

运行结果:

请输入 4 个数: 6,8,1,4 ✓
a=6,b=8,c=1,d=4
排序结果如下:
4 6 8

3.8 有 4 个圆塔,圆心分别为(2,2)、(-2,2)、(-2,-2)、(2,-2),圆半径为 1m,见

图 3-3。这 4 个塔的高度为 10m, 塔以外无建筑物。现输入任一点的坐标, 求该点的建筑高度(塔外的高度为 0)。

解: N-S 图见图 3-4。

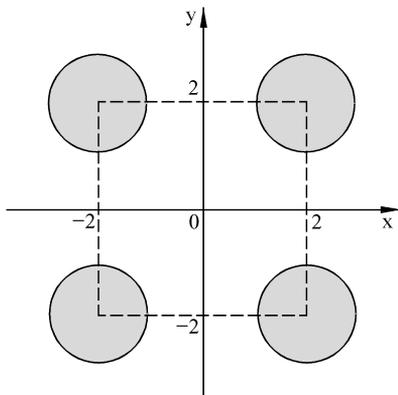


图 3-3

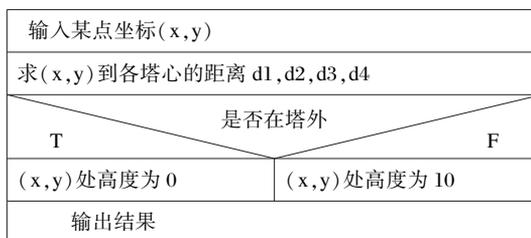


图 3-4

编写程序如下。

```
#include<stdio.h>
int main()
{ int h=10;
  float x=2,y1=2,x2=-2,y2=2,x3=-2,y3=-2,x4=2,y4=-2,x,y,
        d1,d2,d3,d4;
  printf("请输入一个点(x,y):");
  scanf("%f,%f",&x,&y);
  d1=(x-x4)*(x-x4)+(y-y4)*(y-y4);           /* 求该点到各中心点距离 */
  d2=(x-x1)*(x-x1)+(y-y1)*(y-y1);
  d3=(x-x2)*(x-x2)+(y-y2)*(y-y2);
  d4=(x-x3)*(x-x3)+(y-y3)*(y-y3);
  if (d1>1 && d2>1 && d3>1 && d4>1) h=0;    /* 判断该点是否在塔外 */
  printf("该点高度为 %d\n",h);
  return 0;
}
```

运行结果:

- ① 请输入一个点(x,y): 0.5,0.7 ✓
该点高度为 0
- ② 请输入一个点(x,y): 2.1,2.3 ✓
该点高度为 10