

判断程序运行结果

为了模拟试卷风格，例题的代码风格都是试卷上的风格，肯定不是我懒不愿意改--_--。

前置++/后置++/逗号表达式

前置++是先自身+1，再将+1后的值返回

后置++是先返回原来的值，再自身+1

逗号表达式：从左到右逐个计算，最后表达式的结果是最后一个表达式的结果：如

```
1  int a=3,b=5;
2  a=(a++,b=a,a+3);
```

那么

```
1  a=7,b=4
```

看例题

```
1  #include<stdio.h>
2  main()
3  {  int a=1,b=2,c=3;
4      c=(++c,b--,a++);
5      printf("a=%d,b=%d,c=%d\n",a,b,c);
6  }
```

那么结果就是

```
1  a=2,b=1,c=1
```

for循环/while循环/do-while循环

for循环

```
1  for(循环变量赋初值;循环条件;循环变量增值){
2      循环体
3  }
```

```
1  for(int i=1;i<=n;i++){
2      printf("%d\n",i);
3  }
```

while循环

```
1  while(循环条件){
2      循环体
3  }
```

```
1  int i=1;
2  while(i<=n){
3      printf("%d\n",i);
4      i++;
5  }
```

do-while循环

```
1  do{
2      循环体
3  }while(循环条件);
```

```
1  int i=1;
2  do{
3      printf("%d\n",i);
4      i++;
5  }while(i<=n);
```

看例题

```
1  main()
2  {
3      int n, sum=0;
4      for(n=1;n<=20;n++)
5          sum=sum+2;
6      printf("result=%d\n",sum);
7  }
```

运行结果

```
1  result=40
```

```
1  #include<stdio.h>
2  void main()
3  {
4      int k=0;
5      while(k<=2){
6          k++;
7          printf("%d\n",k);
8      }
9  }
```

运行结果

```
1  1
2  2
3  3
```

```

1  #include <stdio.h>
2  main()
3  {
4      int i=1,j=10,sum=0;
5      do{
6          sum=sum+i+j;
7          i++;j--;
8      }while(i<=j);
9      printf("sum=%d\n",sum);
10 }

```

运行过程

```

1  i=1,j=10,sum=0
2  i=2,j=9,sum=11
3  i=3,j=8,sum=22
4  i=4,j=7,sum=33
5  i=5,j=6,sum=44
6  i=6,j=5,sum=55

```

运行结果

```

1  sum=55

```

break/continue

break 跳出循环

continue 跳出本次循环进入下一次循环

例题

```

1  #include <stdio.h>
2  void main()
3  {
4      int a,b;
5      for(a=1,b=1;a<=100;a++)
6      {
7          if(b>=20) break;
8          if(b%3==1){b+=3;continue;}
9          b=b-5;
10     }
11     printf("%d\n",a);
12 }

```

运行过程

```

1  a=1,b=1
2  a=2,b=4
3  a=3,b=7
4  a=4,b=10
5  a=5,b=13
6  a=6,b=16
7  a=7,b=19
8  a=8,b=21

```

运行结果

```
1 | a=8
```

函数/数组/字符数组/指针

函数

传值：在函数里改变不改变函数外的原值，相当于在函数里做了形参的临时拷贝

传址：在函数里改变改变函数外的原值，在函数外用 `&()` 传参，在函数里用 `*` 获取原值

```
1 void fun(int m,int *n){ //m是传值，n是传址
2     m=2;
3     *n=2;
4 }
5 int main(){
6     int a=1,b=1;
7     fun(a,&b);
8     return 0;
9 }
```

数组

数组下标从 0 开始

数组名是首元素地址，数组传参的时候传数组名相当于传址，可以用 `a[]` 接收，也可用指针接收

由于传的是地址，那么函数外的值也会改变

```
1 void fun(int x[],int y[],int *z){
2     printf("%d %d %d %d",x[1],y[1],z[1],*(z+1));
3 }
4 int main(){
5     int a[3]={1,2,3};
6     int b[3]={4,5,6};
7     fun(a,a+1,b);
8     return 0;
9 }
```

字符数组

存字符的数组，不过不同的是字符数组末尾多一个 `\0`

可以用 `gets(a)` 输入一行字符到 `a`（包括空格）【C++14后 `gets()` 函数被完全删除，可以用 `getline(cin,a)` 替代】

可以用 `puts(a)` 将 `a` 输出到屏幕，并加上回车

指针

指针存地址，用 `*` 调用原值，一般用与取地址元素相同类型的指针存地址

例题

```

1  #include <stdio.h>
2  int func(int b[],int n)
3  {   int i,s;
4      s=0;
5      for (i=0;i<n;i++)  s=s+b[i];
6      return (s);
7  }
8  void main()
9  {   int m,b[]={9,8,7,6,5,4,3,2,1};
10     m=func(b+2,3);
11     printf("m=%d\n",m);
12 }
13

```

运行结果

```

1  | m=18

```

```

1  #include<stdio.h>
2  main()
3  {
4  char a[]="ABCDE",*p;
5  int i=0;
6  p=a;
7  while(*p!='\0')
8      {   if( i%2 ==0 ) *p = '*';
9          p++ ;  i++;
10      }
11  puts(a);
12 }

```

运行结果

```

1  | *B*D*

```

程序填空

1、翻转数组

以下是一个翻转数组的程序，根据提示填上合适的条件使其能够正确执行。

```

1  #include<stdio.h>
2  #define M 8
3  main()
4  {   int a[M],i,j,t;
5      for(i=0;i<M;i++)
6          scanf("%d",a+i);
7      i=0;j=M-1;
8      while(i<j)
9      {   t=*(a+i);_____;_____=t;
10         i++;   j--;
11     }
12     for(i=0;i<M;i++)   printf("%3d",*(a+i));
13 }

```

答案

```

1  *(a+i)=*(a+j)
2  *(a+j)

```

2、排序

以下是一个利用选择排序法对长度为 n 的一维整型数组进行升序排序的函数，将函数中的空白处填上合适的语句，使其完整。

```

1  void SelectionSort(int a[], int n)
2  {
3      int i,j,temp;
4      for (j=0;j<n-1;j++)
5      {
6          for (i=0;i<n-1-j;i++)
7              if (_____)
8              {
9                  temp = _____;
10                 a[i] = a[i+1];
11                 a[i+1] = _____;
12             }
13     }
14 }

```

答案

```

1  a[i]>a[i+1]
2  a[i]
3  temp

```

3、统计字符

以下是一个统计字符串中各种字符的程序，根据提示填上合适的条件使其能够正确执行。

```

1  #include <stdio.h>
2  #define N 80
3  int main()
4  {
5      char str[N];

```

```

6   int i, letter = 0, digit = 0, space = 0, others = 0;
7   printf("Input a string:");
8   gets(str);
9   for (i=0; str[i]!='\0'; i++)
10  {
11      if (_____)
12          letter ++;          /* 统计英文字符 */
13      else if (_____)
14          digit ++;          /* 统计数字字符 */
15      else if (_____)
16          space ++;          /* 统计空格 */
17      else
18          others ++;          /* 统计其他字符 */
19  }
20  printf("English character: %d\n", letter);
21  printf("digit character: %d\n", digit);
22  printf("space: %d\n", space);
23  printf("other character: %d\n", others);
24  return 0;
25  }

```

答案

```

1   (str[i]>='a'&&str[i]<='z')||(str[i]>='A'&&str[i]<='Z')
2   str[i]>='0'&&str[i]<='9'
3   str[i]==' '

```

4、指针访问数组

以下是一个利用指针访问数组中元素的程序，在空白处填上合适的语句，使其能正确执行。

```

1   #include<stdio.h>
2   int main()
3   {
4       int a[5]={1,2,3,4,5}, *p,i=0;
5       _____;
6       for (i=0;i<5;i++)
7       {
8           printf("%d ", *(p+i));
9       }
10      printf("\n");
11      return 0;
12  }

```

答案

```

1   p=a

```

5、链表

以下是一个将一个结点插入到链表中已知结点的后面的函数，将函数补充完整使其能够正确执行。

```

1   struct node
2   {   int data;
3       struct node *next;

```

```

4  };
5  struct node* insert(struct node* head,int data,int n)
6  /*n为待插入节点的值，data为已知节点的值，n所在节点插入到已知data的后面*/
7  {
8      struct node *p,*s;
9      p=head;
10     while(p!=NULL)
11     {
12         if(p->data==data)break;
13         _____;
14     }
15     s=( struct node*)malloc(sizeof(struct node));
16     _____;
17     _____;
18     return head;
19 }

```

答案

```

1  p=p->next
2  s->next=p->next
3  p->next =s

```

下程序段的功能是统计单向链表中所有节点中数据的和，其中 head 为指向第一个节点的指针。请填空。

```

1  struct link
2  { int data;
3    struct link *next;
4  };
5  int sum(_____)
6  { struct link *p;
7    int c=0;
8    p=head;
9    while(_____)
10   {
11     c=c+p->data;
12     p=_____;
13   }
14   return c;
15 }

```

答案

```

1  struct link *head
2  p!=NULL
3  p->next

```

6、判断闰年

已知能被 4 整除而不能被 100 整除的或者能被 400 整除的年份是闰年，则判断某一年是否是闰年的程序如下：

```
1  main()
2  {
3      int year, leap;
4      scanf("%d", &year);
5      if(_____)
6          leap=1;
7      else leap=0;
8      if(_____) printf("是闰年");
9      else printf("不是闰年");
10 }
```

答案

```
1  (year%4==0&&year%100!=0) || (year%400==0)
2  leap==1
```

7、数学题

计算 $1 - 1/2 + 1/3 - 1/4 + \dots + 1/99 - 1/100 + \dots$,直到最后一项的绝对值小于 10^{-4} 为止。

```
1  #include<stdio.h>
2  #include<math.h>
3  void main()
4  {   int n=1;
5      float term=1.0, sign=1, sum=_____;
6      while(_____)
7      {
8          _____;
9          sum=sum+term;
10         sign=_____;
11         n++;
12     }
13     printf("sum=%f\n", sum);
14 }
```

答案

```
1  0
2  fabs(term)>=1e-4
3  term=sign/n
4  -sign
```

程序改错

1、数组越界/访问数组方式错误

```
1 #include<stdio.h>
2 int main()
3 {
4     int a[]={2,4,6,8,10};
5     int i;
6     for(i=1;i<=5;i++)
7         printf("%d ",*a++);
8     return 0;
9 }
```

改错

```
1 for(i=1;i<=5;i++)    --->    for(i=0;i<5;i++)
2 printf("%d ",*a++);  --->
3     printf("%d ",*(a+i)); 或改成printf("%d ",a[i]);
```

2、函数参数类型不匹配

```
1 #include <stdio.h>
2 void S(int *x, int *y);
3 int main()
4 {
5     int a=10, b=20;
6     printf("a = %d, b = %d\n", a, b);
7     S(a, b);
8     printf("a = %d, b = %d\n", a, b);
9     return 0;
10 }
11 void S(int *x, int *y)
12 {
13     int temp;
14     temp = *x;
15     *x = *y;
16     *y = temp;
17 }
```

改错

```
1 S(a, b);    --->    S(&a, &b);
```

3、打印格式化数据类型/数量不匹配

```
1 #include <stdio.h>
2 void main()
3 { int a[4]={1,3,5,7};
4     printf("%d%d%d%d",a);
5 }
```

改错

```
1 printf("%d%d%d%d",a);    --->
2     printf("%d%d%d%d",a[0],a[1],a[2],a[3]);
```

4、if使用单等号

```
1 #include <stdio.h>
2 int main()
3 {
4     int a=3; a++;
5     if(a=3) puts("yes");
6     else puts("no");
7     return 0;
8 }
```

改错

```
1 if(a=3) puts("yes");    --->    if(a==3) puts("yes");
```

5、输入没加取地址符号

```
1 #include <stdio.h>
2 int main()
3 {
4     int a;
5     scanf("%d",a);
6     return 0;
7 }
```

改错

```
1 scanf("%d",a);    --->    scanf("%d",&a);
```

程序设计

1、求最小值

编写一个程序，计算长度为 10 的一维整型数组中最小数并输出，要求数组中的值从键盘输入。

```
1 #include<stdio.h>
2 int main()
3 {
4     int a[10],i,mi;
5     for(i=0;i<10;i++)
6         scanf("%d",&a[i]);
7     mi=a[0];
8     for(i=1;i<10;i++)
9         if(a[i]<mi)
10             mi=a[i];
11     printf("最小值为%d",mi);
12     return 0;
13 }
```

2、求素数

编写程序输出 100 以内的所有素数。

```
1  #include<stdio.h>
2  int main()
3  {
4      int i,j;
5      for(i=2;i<=100;i++)
6      {
7          for(j=2;j<i;j++)
8              if(i%j==0)
9                  break;
10         if(i==j)
11             printf("%d ",i);
12     }
13     return 0;
14 }
```

3、矩阵转置

输入一个 3×3 矩阵然后转置后输出。

```
1  #include<stdio.h>
2  int main()
3  {
4      int i,j;
5      int a[3][3];
6      for(i=0;i<3;i++)
7          for(j=0;j<3;j++)
8              scanf("%d",&a[i][j]);
9      for(i=0;i<3;i++)
10     {
11         for(j=0;j<3;j++)
12             printf("%d ",a[j][i]);
13         printf("\n");
14     }
15     return 0;
16 }
```

4、逆置数组

在主函数中输入 n 个整数，并输出，之后调用逆置函数将 n 个数逆置，并在主函数中输出逆置后的结果。

```
1  #include<stdio.h>
2  void reverse(int a[],int n)
3  {
4      int i,j,t;
5      for(i=0,j=n-1;i<j;i++,j--)
6      {
7          t=a[i];
8          a[i]=a[j];
9          a[j]=t;
10     }
11 }
```

```

12 int main()
13 {
14     int i,n,a[100];
15     printf("输入数的个数");
16     scanf("%d",&n);
17     for(i=0;i<n;i++)
18         scanf("%d",&a[i]);
19     reverse(a,n);
20     for(i=0;i<n;i++)
21         printf("%d ",a[i]);
22     return 0;
23 }

```

5、字符串连接

编写一个函数实现两个字符串的连接并在主函数中调用该函数。（要求不能使用系统的函数）

```

1  #include<stdio.h>
2  void myStrcat(char s1[],char s2[])
3  {
4      while(*s1!='\0') s1++;
5      while((*s1++)=*(s2++))!='\0');
6  }
7  int main()
8  {
9      char s1[100],s2[100];
10     scanf("%s%s",s1,s2);
11     myStrcat(s1,s2);
12     puts(s1);
13     return 0;
14 }

```

6、字符串比较

编写一个函数实现两个字符串的比较并在主函数中调用该函数。（要求不能使用系统库函数！）

```

1  #include<stdio.h>
2  int myStrcmp(char s1[],char s2[])
3  {
4      while(*s1==*s2&&*s1!='\0'&&*s2!='\0')
5          s1++,s2++;
6      return *s1-*s2;
7  }
8  int main()
9  {
10     char s1[100],s2[100];
11     scanf("%s%s",s1,s2);
12     printf("%d",myStrcmp(s1,s2));
13     return 0;
14 }

```

7、文件大题（默写题）

定义一个包含学号、姓名、性别和通信地址基本信息的结构体，利用该结构体实现

(1) 将 5 个学生的基本信息写入磁盘文件中。

(2) 将磁盘文件中的信息读出并在屏幕上显示。

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  typedef struct student
4  {
5      int id;
6      char name[100];
7      char sex[10];
8      char add[100];
9  }STUDENT;
10 void input(STUDENT stu[],int n);
11 void output(STUDENT stu[],int n);
12 void writefile(STUDENT stu[],int n);
13 void readfile(STUDENT stu[],int n);
14
15 int main()
16 {
17     STUDENT stu1[5],stu2[5];
18     input(stu1,5);
19     writefile(stu1,5);
20     readfile(stu2,5);
21     output(stu2,5);
22     return 0;
23 }
24 void input(STUDENT stu[],int n)
25 {
26     int i,j;
27     for(i=0;i<5;i++)
28     {
29         scanf("%d",&stu[i].id);
30         scanf("%s",stu[i].name);
31         scanf("%s",stu[i].sex);
32         scanf("%s",stu[i].add);
33     }
34 }
35 void output(STUDENT stu[],int n)
36 {
37     int i;
38     for(i=0;i<5;i++)
39     {
40         printf("%d ",stu[i].id);
41         printf("%s ",stu[i].name);
42         printf("%s ",stu[i].sex);
43         printf("%s\n",stu[i].add);
44     }
45 }
46 void writefile(STUDENT stu[],int n)
47 {
48     FILE *fp;
49     fp=fopen("a.txt","w");
50     if(fp==NULL)
```

```

51     {
52         printf("打开失败\n");
53         exit(0);
54     }
55     fwrite(stu,sizeof(STUDENT),5,fp);
56     fclose(fp);
57 }
58 void readfile(STUDENT stu[],int n)
59 {
60     int i;
61     FILE *fp;
62     fp=fopen("a.txt","r");
63     if(fp==NULL)
64     {
65         printf("打开失败\n");
66         exit(0);
67     }
68     fwrite(stu,sizeof(STUDENT),5,fp);
69     for(i=0;!feof(fp);i++)
70     {
71         fread(&stu[i],sizeof(STUDENT),1,fp);
72     }
73     fclose(fp);
74 }

```

祝大家考的全会，蒙的全对，补考顺利!! (σ'ω')σ

主讲：张一鸣