

## 递归函数

### 1. 单选

- 以下程序的输出结果是：\_\_\_\_\_。

```
#include <stdio.h>
int fib(int n)
{
    if(n>2) return(fib(n-1)+fib(n-2));
    else return(2);
}
int main()
{
    printf("%d", fib(3));
    return 0;
}
```

- A) 2
- B) 4
- C) 6
- D) 8

- 以下程序的输出结果是：\_\_\_\_\_。

```
#include <stdio.h>
int f(int n)
{
    if(n==1) return 1;
    else return f(n-1)+1;
}
int main()
{
    int i, j=0;
    for(i=1; i<3; i++) j+=f(i);
    printf("%d", j);
    return 0;
}
```

- A) 4
- B) 3
- C) 2
- D) 1

- 以下程序的输出结果是：\_\_\_\_\_。

```
#include <stdio.h>
void fun(int *a, int i, int j)
{

```

```

int t;
if(i<j)
{
    t=a[i]; a[i]=a[j]; a[j]=t;
    fun(a, ++i, --j);
}
}
int main()
{
    int a[]={2, 4, 1, 3, 6, 5}, i;
    fun(a, 0, 5);
    for(i=0; i<6; i++) printf("%d", a[i]);
    return 0;
}

```

A)563142

B)654321

C)123456

D)365124

- 下面程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
int age(int n)
{
    int c;
    if(n==1)
        c=10;
    else
        c=age(n-1)+2;
    return c;
}
int main()
{
    printf("%d\n",age(5));
    return 0;
}

```

A) 14

B) 16

C) 18

D) 20

- 下面程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
int fac(int n)
{
    int fact;
    if(0==n || 1==n)

```

```

        fact=1;
    else
        fact=fac(n-1)*n;
    return fact;
}
int main()
{
    printf("%d\n",fac(5));
    return 0;
}

```

- A) 12
- B) 15
- C) 120
- D) 150

- 下面程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
int fac(int n)
{
    int fact;
    if(0==n || 1==n)
        fact=1;
    else
        fact=fac(n-1)+n;
    return fact;
}
int main()
{
    printf("%d\n",fac(5));
    return 0;
}

```

- A) 12
- B) 15
- C) 120
- D) 150

## 2. 程序运行结果

- 以下程序的输出结果是：\_\_\_\_\_。

```

#include <stdio.h>
int f(int a[], int n)
{
    if(n>1)
        return(a[0]+f(&a[1], n-1));
    else

```

```

        return(a[0]);
    }
int main()
{
    int a[3]={1, 2, 3}, s;
    s=f(&a[0], 3);
    printf("%d\n", s);
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
void fun(char *s)
{
    char t;
    if(*s != '\0')
    {
        t = *s++;
        fun(s);
        if(t != '\0') printf("%c", t);
    }
}
int main()
{
    char *a="1234";
    fun(a);
    printf("\n");
    return 0;
}

```

- 分析以下程序的执行结果：

```

#include <stdio.h>
int foo(int x, int y)
{
    if(x<=0 || y<=0) return 1;
    return 3*foo(x-1, y/2);
}
int main()
{
    printf("%d\n", foo(3, 5));
    return 0;
}

```

### 3. 程序完形填空

- 若程序的运行结果如下：

```
fact(5): 120
```

```
fact(1): 1
```

```
fact(<0): error
```

请填空使程序完整:

```
#include <stdio.h>
```

```
int fact( (1) )
```

```
{
```

```
    if(value<0)
```

```
    {
```

```
        printf(" (2) \n");
```

```
        return -1;
```

```
    }
```

```
    else if(value == 1)
```

```
        return 1;
```

```
    else
```

```
        return (3) ;
```

```
}
```

```
int main()
```

```
{
```

```
    printf("fact(5): %d\n", fact(5));
```

```
    printf("fact(1): %d\n", fact(1));
```

```
    fact(-5);
```

```
    return 0;
```

```
}
```