

填空题

1.下面程序段的运行结果是

【正确答案: 600】

```
char ch[ ] = "600";  
int a, s = 0;  
for (a = 0; ch[a] >= '0' && ch[a] <= '9'; a++)  
    s = 10 * s + ch[a] - '0';  
printf("%d", s);
```

2.写出下述程序的输出结果

【正确答案: 58】

```
#include <stdio.h>  
int func(int n)  
{  
    if(n<=1)  
        return 1;  
    else  
        return (2+n*func(n-1));  
}
```

```
int main()  
{  
    int x = 4;  
    printf("%d\n",func(x));  
  
    return 0;  
}
```

3.给出下述程序的执行结果

【正确答案: 8】

```
#include <stdio.h>  
long fib(int x)  
{  
    switch(x)  
    {  
        case 0: return 0;  
        case 1:  
        case 2: return 1;  
    }  
    return (fib(x-1)+fib(x-2));  
}  
int main()  
{  
    int x=6;  
    printf("%d\n",fib(x));  
}
```

```
return 0;
}
```

4. 下面的函数 invert 的功能是将一个字符串的内容颠倒过来。请将函数补充完整。

【正确答案: k -1】(所有答案填写在此空内，答案间以空格隔开)

```
void invert(char str[])
{
    int i,j,填空 1_____;
    for(i=0,j=strlen(str)填空 2_____;i<j;i++,j--)
    {
        k=str[i];
        str[i]=str[j];
        str[j]=k;
    }
}
```

5. 下列程序是否正确，如果正确给出输出结果，如果不正确，答案处填写“不正确”。

【正确答案: 不正确】

```
#include <stdio.h>
void func(char *q)
{
    char a[]="hello";
    q=a;
}
main()
{
    char *p;
    func(p);
    printf("%s\n",p);
}
```

6. 函数 index(char s[],char t[])检查字符串 s 中是否包含字符串 t，若包含，则返回 t 在 s 中的开始位置(下标值)，否则返回-1。请将函数补充完整。

```
int index(char s[],char t[])
{
    int i,j,k;
    for(i=0;s[i]!='\0';i++)
    {
        for(j=i,k=0;填空 1_____&s[j]==t[k];j++,k++)
        ;
        if(填空 2_____)
            return i;
    }
    return -1;
}
```

```
}
```

【 正确答案: $t[k] != '\0'$ $t[k] == '\0'$ 或 $s[j] != '\0'$ $t[k] == '\0'$ 或 $t[k] != 0$ $t[k] == 0$ 或 $s[j] != 0$ $t[k] == 0$ 或 $k < \text{strlen}(t)$ $k == \text{strlen}(t)$ 或 $s[j] != '\0' \&\& t[k] != '\0'$ $t[k] == '\0'$ 或 $t[k] != '\0'$ $k == \text{strlen}(t)$ 或 $s[j] != '\0'$ $k == \text{strlen}(t)$ 或 $s[j] != '\0' \&\& t[k] != '\0'$ $k == \text{strlen}(t)$ 或 $t[k] != 0$ $k == \text{strlen}(t)$ 或 $s[j] != 0$ $k == \text{strlen}(t)$ 或 $k < \text{strlen}(t)$ $t[k] == '\0'$ 或 $k < \text{strlen}(t)$ $t[k] == 0$ 或 $t[k] != '\0' \&\& s[j] != '\0'$ $t[k] == '\0'$ 或 $t[k] != '\0' j - i == \text{strlen}(t)$ 】 (所有答案填写在此空内, 答案间以空格隔开)

7. 函数 `squeez(char s[], char c)` 的功能是删除字符串 `s` 中所出现的与变量 `c` 相同的字符。

```
void squeez(char s[], char c)
{
    int i, j;
    for(i=j=0; 填空 1 _____; i++)
        if(s[i] != c)
            填空 2 _____;
        s[j++] = '\0';
}
```

【 正确答案: $s[i] != '\0'$ $s[j++] = s[i]$ 或 $i < \text{strlen}(s)$ $s[j++] = s[i]$ 或 $s[i] != 0$ $s[j++] = s[i]$ 或 $i < \text{strlen}(s) - 1$ $s[j++] = s[i]$ 】 (所有答案填写在此空内, 答案间以空格隔开)

8. 下面的函数 `itoh(n, s)` 完成将无符号十进制整数转换成十六进制表示, 并存入字符串数组 `s` 中。程序中用到的函数 `reverse(char s[])` 是一个将字符串置逆的函数。

```
void reverse(char s[])
{
    int temp, i, j;
    for(i=0, j=strlen(s)-1; i < j; i++, j--)
    {
        temp = s[i];
        s[i] = s[j];
        s[j] = temp;
    }
}

void itoh(unsigned n, char s[])
{
    int h, i=0;
    do{
        h = n%16;
        s[i++] = (h <= 9)? h + '0': h + 'A' _____;
    }while((n/=16) != 0);
    _____;
    reverse(s);
}
```

```
}
```

【正确答案: -10 s[i]='\0' 或 -10 s[i]=0】(所有答案填写在此空内, 答案间以空格隔开)

9.下面程序的功能是将字符串 s 中的内容按逆序输出,但不改变串中的内容,请选择填空(答案请填写所选择的两个字母,中间用空格隔开,如 A A 的形式):

【正确答案: C B】

```
#include <stdio.h>
void inverp(char *a)
{
    if ( _____ )
        return;
    inverp(a+1);
    printf("%c", _____ );
}
int main()
{
    char s[10] = "hello!";
    inverp(s);

    return 0;
}
```

Selection 1 :

- (A) *a != '\0'
- (B) *a != NULL
- (C) *a == '\0'
- (D) !a * == 0

Selection 2 :

- (A) *(a-1)
- (B) *a
- (C) *(a+1)
- (D) * (--a)

10. 下面程序的功能是将已按升序排好的两个字符串 a 和 b 中的字符按升序并归到字符串 c 中。请选择填空将程序补充完整:(所有答案填写在此空内, 答案间以空格隔开)

【正确答案: A D A】

```
#include<stdio.h>
#include<string.h>
int main()
{
    char a[]="acegikm";
    char b[]="bdfhjlnpq";
```

```
char c[80],*p;  
int i=0,j=0,k=0;  
while(a[i]!='\0'&&b[j]!='\0')  
{  
    if(a[i]<b[j]){  
        else{  
    }  
    c[k]='\0';  
    if( )p=b+j;  
    else p=a+i;  
    strcat(c,p);  
    puts(c);  
  
    return 0;  
}
```

填空 1:

- (A)c[k++]=a[i++];
- (B)c[k++]=a[j++];
- (C)c[k++]=b[i++];
- (D)c[k++]=b[j++];

填空 2:

- (A)c[k++]=a[i++];
- (B)c[k++]=a[j++];
- (C)c[k++]=b[i++];
- (D)c[k++]=b[j++];

填空 3:

- (A)a[i]=''\0'
- (B)a[j]=''\0'
- (C)b[i]=''\0'
- (D)b[j]=''\0'