

## 演習 1

次のプログラムの出力結果を、右に記入せよ

```
#include <stdio.h>
#define MAXNUM 10

int main()
{
    int a[MAXNUM];
    int *p1, *p2;
    int i;

    p1=a;
    p2=&a[0];
    for(i=0; i<MAXNUM; i++)
    {
        *p1=i*2;
        p1++;
    }
    *p2=10;
    for(i=0; i<MAXNUM; i++)
    {
        printf("%d ", a[i]);
    }

    return 0;
}
```

## 演習 2

次のプログラム(左のstrEdit関数と右のmain関数)の出力結果を、下に記入せよ

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

```
void strEdit(char *str)
```

```
{
    int i, j, cnt;
    char temp;

    for( cnt = 0 ; *( str + cnt ) != '¥0' ; cnt++ ) ;
    for( i = 0 ; i < cnt - 1 ; i++ )
    {
        for( j = i + 1 ; j < cnt ; j++ )
        {
            if( *(str + i) > *(str + j) )
            {
                temp=*(str + i);
                *(str + i)=*(str + j);
                *(str + j)=temp;
            }
        }
    }
}
```

```
int main()
```

```
{
    char str []="METALGEARSOLID";

    strEdit(str);
    printf("%s¥n", str);
    return 0;
}
```

### 演習 3

次のプログラムの出力結果を、右に記入せよ

```
#include <stdio.h>

void strEdit(char *);

int main()
{
    char str[3][10]={"Windows",
                    "Mac",
                    "Linux"};

    int i;

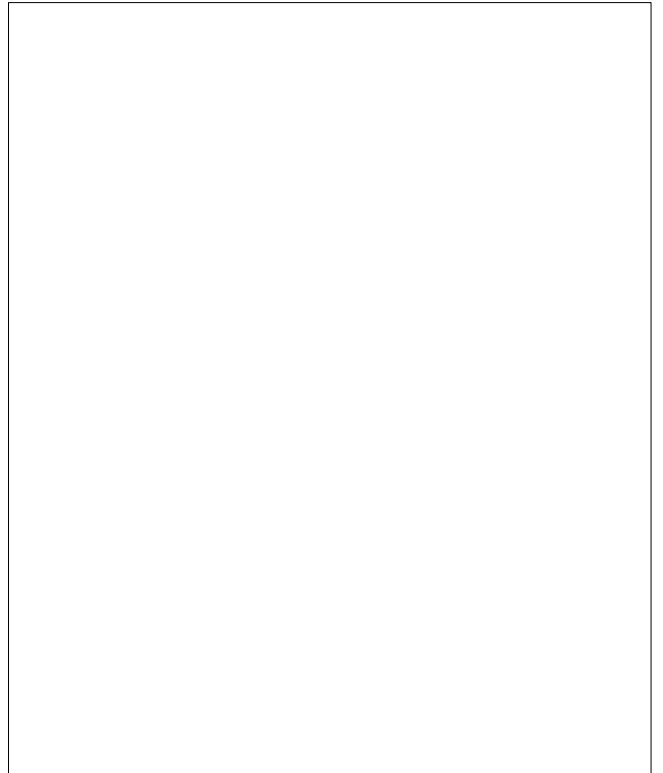
    for(i=0;i<3;i++)
    {
        strEdit(&str[i][0]);
    }
    for(i=0;i<3;i++)
    {
        printf("%s¥n", &str[i][0]);
    }

    return 0;
}
```

```
void strEdit(char *str)
{
    char *p1,*p2,temp[10];

    p1=str;
    p2=temp;
    while( *p1 != '¥0' )
    {
        p1++;
    }
    while(p1 != str)
    {
        p1--;
        *p2=*p1;
        p2++;
    }
    *p2=' ¥0' ;

    p1=str;
    p2=temp;
    while(*p2 != ' ¥0' )
    {
        *p1=*p2;
        p1++;p2++;
    }
}
```



## 演習 4

次のプログラムは、ある文字列 (str1) に対し、その文字列を正の整数値へ変換する関数と、その動作確認用コードである。空欄をうめて、正しく動作するようにせよ。

なお、入力する文字列は数字のみによって構成されるとし、マイナスや文字を含んでいる場合は考慮しなくて良い。

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

```
int strEdit(char *str)
{
    int num=0;

    while(①_____ )
    {
        num*=10;
        num+=②_____ ;
        str++;
    }
    return ③_____ ;
}
```

```
int main()
{
    char str[11];
    int strNum;

    printf("数字を入力=>");
    scanf("%10s", str);
    strNum=strEdit(str);
    printf("値は%d¥n", strNum);

    return 0;
}
```

## 演習 5

演習 4 のプログラムを、マイナスの値に対応できるようにしたい。

どこにどのようなコードを追加すればよいか。場所と処理を説明する、あるいはコードで表せ。

場所

処理

## 演習 5

特定の文字列が連続（AAAAAのような場合）し、3個を超えるならば、「アルファベット\*個数」と言う書き方、具体的に言えば「A\*5」と言った書き方へ変え、新しく画面に出力する。このプログラムを以下に示すので、空欄をうめよ。なお、連続する文字は9個以下とする。

```
#include <stdio.h>
void strEdit(char *str)
{
    char ①_____ ;
    char temp[1000];
    int i, ct;

    p1=p3=str;
    p2=temp;
    while( *p3 != '\0' )
    {
        ct=0;
        do { p1++; ②_____ ; } while( *p1 == *p3 );
        if( ct > 3 )
        {
            *p2=*p3;    p2++;
            *p2='*';    p2++;
            *p2=ct+'0'; p2++;
        }
        else
        {
            for(i=0; i<ct; i++)
            {
                ③_____ ; p2++;
            }
        }
        ④_____ ;
    }
    *p2='\0' ;
    i=0;
    while(temp[i] != '\0' )
    {
        *str=temp[i];
        str++;    i++;
    }
    *str=temp[i];
    printf("%s\n", temp);
}

int main()
{
    char str[1000]="aaaasfgeaghaegaahiiisksiiiiis";

    printf("%s\n", str);
    strEdit(str);
    printf("%s\n", str);
    return 0;
}
```

## 演習 6

インベーターを自分で作ってみようとしていると仮定し、自機に関するデータをPLAYER構造体で保持するとする。この構造体にはどのようなメンバが必要だろうか？検討してみよ。

```
struct PLAYER {
```

```
};
```

## 演習 7

では、スマブラの操作キャラクタであればどうか？  
同じPLAYER型を4人のプレイヤーないしCOMが使い、内容で区別するとして検討してみよ。

```
struct PLAYER {
```

```
};
```

### 演習 1 解答

```
10 2 4 6 8 10 12 14 16 18
```

### 演習 2 解答

```
AADEEGILLMORST
```

### 演習 3 解答

文字を逆順に入れ替えているだけである。

```
swodniW  
caM  
xuniL
```

### 演習 4 解答

- ① \*str!=' ¥0'
- ② \*str-' 0'
- ③ num

### 演習 5 解答

場所

空欄①のwhile文の前

処理の変更・追加内容

一文字目がマイナスか、数字ないしプラスで条件分岐し、マイナスの時だけフラグを立てる。  
その後、returnで返す時にマイナスの時の符号を掛けることにする。

### 演習 5 解答

- ① \*p1, \*p2, \*p3
- ② ct++
- ③ \*p2=\*p3
- ④ p3 = p1

### 演習 6 解答「例」

```
struct PLAYER{  
    int posX;  
    int score;  
    int body;  
};
```

### 演習 7 解答「例」

```
struct PLAYER{  
    int posX, posY;  
    int charaNumber;  
    int score;  
    int body;  
};
```

### 演習 6 演習 7 について

これら以外に、前の技を発動したCPU時間など、まだまだ必要な物は考えられるので、いろいろ考えてみるとよい。