

for 文, while 文:解答

◆ A : 練習問題

1. (`・ω・´) シヤキン を 10 行×10 列で出力するプログラム

```
#include<stdio.h>
int main(void){
    int i,j;
    for(i=0;i<10;i++){
        for(j=0;j<10;j++){
            printf("( `・ω・´ ) シヤキン");
        }
        printf("\n");
    }
    return(0);
}
```

2. 7 の倍数が入力されるまでループするプログラム

```
#include<stdio.h>
int main(void){
    int a;
    while(1){
        printf("7の倍数を入力せよ。");
        scanf("%d",&a);
        if(a%7==0){
            printf("入力されました。");
            break;
        }
    }
    return(0);
}
```

`while(1)`とすることで、無限ループが起きます。そこで `break;` を使用することにより、一定の条件を満たしたとき（ここでは 7 の倍数が入力されたとき）、その無限ループを抜ける、という形にします。

3. 2重ループで九九を表示するプログラム

▽追加問題

1. 1×1 から 9×9 までの九九を出力せよ。ただ、文字がズレないように桁を合わせてして出力すること。（ヒント：▽第一回を参考）

```
#include<stdio.h>
int main(void){
    int i,j;
    printf("九九の表！\n");
    for(i=1;i<=9;i++){
        for(j=1;j<=9;j++){
            printf("%3d",i*j);
        }
        printf("\n");
    }
    return(0);
}
```

九九の結果は $i*j$ 。 i の値が変わる = 次の段へ移るとき改行を挟むようにします。また出力を `%3d` とすることにより、出力はすべて 3 ケタになります。

★出力結果

```
九九の表！
 1  2  3  4  5  6  7  8  9
 2  4  6  8 10 12 14 16 18
 3  6  9 12 15 18 21 24 27
 4  8 12 16 20 24 28 32 36
 5 10 15 20 25 30 35 40 45
 6 12 18 24 30 36 42 48 54
 7 14 21 28 35 42 49 56 63
 8 16 24 32 40 48 56 64 72
 9 18 27 36 45 54 63 72 81
```

揃ってはいませんがコマンドプロンプト上では揃って表示されています。

◆ ∇ : 練習問題

1. 上を参照
2. 1 から 1000 までの数のうち、7の倍数か13の倍数である数を出力せよ。

```
#include<stdio.h>
int main(void){
    int i;
    for(i=1;i<=1000;i++){
        if(i%7==0 | i%13==0){
            printf("%5d¥n",i);
        }
    }
    return(0);
}
```

if文の条件は「7で割った時、余りが出ないor13で割った時余りが出ない」となっています。

3. 整数 n を読み込み、1 から n までの二乗の全ての和を出力せよ。
($1^2+2^2+3^2+\dots+n^2$)

```
#include<stdio.h>
int main(void){
    int a,i,sum;
    sum=0;
    scanf("%d",&a);
    for(i=1;i<=a;i++){
        sum=sum+i*i;
    }
    printf("1 からの二乗の和は%d です。",sum);
    return(0);
}
```

まず、入力された数を入れるための変数 a 、for 文の際に使う変数 i 、合計値を入れるための変数 sum を宣言します。

数が入力された後は for 文を使います。合計値は順にそれ用の変数に足していくことで求めます。


```
dialy=dialy%7;
for(i=0;i<7;i++){
    if(dialy==1){
        printf("月曜日です");
        break;
    }
    if(dialy==2){
        printf("火曜日です");
        break;
    }
    if(dialy==3){
        printf("水曜日です");
        break;
    }
    if(dialy==4){
        printf("木曜日です");
        break;
    }
    if(dialy==5){
        printf("金曜日です");
        break;
    }
    if(dialy==6){
        printf("土曜日です");
        break;
    }
    if(dialy==0){
        printf("日曜日です");
        break;
    }
}
return(0);
}
```

目標の日までの日数を計算し、それを 7 で割り、その余りを見ることで曜日を計算し、結果を出します。

2. 1995年1月1日、ヒラタくんはお年玉でゲームを買いました。そのゲームの中では現実の時間の流れとは違い、一ヶ月は30日固定、つまり一年は360日しかありません。2014年5月8日、ユモトくんは勝手にそのゲームを起動させました。当然のことながら現実とは日付が大きすぎてしまっています。何年何月何日でしょうか。

```
#include <stdio.h>
int main(void){
    int i, rday=0;
    int year=0;
    int imonth=0;
    for(i=0; i<4; i++){
        rday=rday+366;           //96年、04年、08年、12年の日数
    }
    for(i=0; i<15; i++){       //上の年以外の年の日数
        rday=rday+365;
    }
    for(i=0; i<2; i++){
        rday=rday+31;          //2014年の1、3月の日数
    }
    rday=rday+28;              //2014年の2月の日数
    rday=rday+30;              //2014年の4月の日数
    rday=rday+8;               //2014年の5月8日までの日数

    //printf("%d日 %Yn", rday); //念のための現実時間の経過日数
    while(rday>360){
        rday=rday-360;         //360日ごとに1年足してく
        year++;
    }
    while(rday>30){            //ひと月ごとに月を足してく
        rday=rday-30;
        imonth++;
    }
    printf("%d年 %d月 %d日", 1995+year, imonth, rday);
    return 0;
}
```

基本的には問 1 と同じです。予定の日までの日数を計算、それを 360 以下になるまで 360 を引き続け、あとは残りの余りを 30 以下になるまでを引き続けます。この時結果は 360 を引いた回数が年、30 を引いた回数が月、最後に余った数が日になります。

ちなみにゲームを起動する日を自分で入力するようにすると以下になります。

```
#include <stdio.h>

int main(void){

    int year1 = 1995 ,month = 1, day = 1;
    int year, month, day;//数えるため
    int uruu = 0;//閏年

    //入力

    printf("何年?");
    scanf("%d",&year);
    printf("何月?");
    scanf("%d",&month);
    printf("何日?");
    scanf("%d",&day);

    while(1){

        day=day+1;           //日にちを上げる
        printf("%d 日¥n",day);
        uruu = 0;
        if(year % 4 == 0 && year % 100 != 0 || year % 400 ==
0){//うるう年判定
            uruu = 1;
        }

        if(day == 30){
            if(month == 4 && month == 6 && month == 9
&& month == 11){
                month = month + 1;
```

```

        day = 1;
        }
    }
    if(day == 31){
        if(month == 1 && month == 3 && month == 5
&& month == 7 && month == 8 && month == 10){
            month = month + 1;
            day = 1;
        }
    }
    if(day == 31){
        if(month == 12){
            month = 1;
            year = 1;
            day = 1;
        }
    }
    if(day == 29){//
        if(uruu == 1){
            month = 3;
            day = 1;
        }
    }
    if(day == 28){//
        if(month == 0){
            month = 3;
            day = 1;
        }
    }
    if(year == year1 && month == month1 && day == day1)
    break;
}

printf("%d 日¥n",day); //念のための現実時間
の経過日数

```

```

        while(day>360){
            day=day-360;                //360 日ごとに
1 年足してく
            year++;
        }
        while(day>30){                //ひと月ごと
に月を足してく
            day=day-30;
            month++;
        }
        printf("%d 年 %d 月 %d 日",1995+year,month,day);
        return 0;
    }

```

閏年を考えなければいけないため、多少厄介かなたちになります。基本的には上の問題と同じです。

3. 整数 N を読み込み、その数が素数かどうか判断せよ。素数でない場合はその数の約数をすべて出力せよ。

```

#include<stdio.h>
int main(void){
    int a,i,flag;
    printf("数を入力してください。");
    scanf("%d",&a);
    if(a<=1){
        printf("この数は素数ではありません。");
        flag=0;
    }
    else{
        for(i=2;i<a;i++){
            if(a%i==0){
                if(flag!=0){
                    printf("この数は素数ではありません。約
数は以下の数です。");
                    flag=0;
                }
            }
        }
        printf("¥n%d",i);
    }
}

```

```
        }  
    }  
}  
if(flag!=0){  
    printf("この数は素数です。");  
}  
return(0);  
}
```

入力された数とその入力された数より小さい数すべてで割り、割れなかった場合素数と判断します。割れた場合は素数でないと判断し、割れた数を出力します。

その判断として変数 **flag** を使います。割り切れた＝素数でない場合は **flag** に 0 を代入、その後は **flag** の数が 0 か 0 で無いかで素数かどうか判断します。