

解答

解答とは言っても一例にすぎません。若干異なっていたとしてもプログラムが動きさえすれば問題ありません。

・ソフトゼミ A 練習問題解答

1. 円の直径の長さを引数として受け取って、その面積を返す関数 ただし円周率は 3.14 としてよい。

```
double Circle(double d)
```

引数が「直径」であることに注意しましょう。半径ではありません。計算した面積の値は実数になるので、関数の型（返り値の型）は double 型です。

```
//円の直径の値を受け取って円の面積の値を返す関数
#include<stdio.h>

double Circle(double d); //関数プロトタイプ宣言

int main(void){
    double d,s;
    scanf("%lf",&d); //scanf で double 型を代入するときは%f ではなく%lf です
    s=Circle(d); //関数 Circle で返ってきた円の面積の値を s に代入
    printf("直径%f の円の面積は%f です\n",d,s);
    return 0;
}

double Circle(double d){
    double x,r;
    r=d/2; //半径
    x=r*r*3.14; //面積の計算
    return x; //面積 x を返す
}
```

2. 整数の値を 3 つ、引数として受け取ってそのうち最大の値を返す関数

```
int max(int a,int b,int c)
```

3 つの中からの最大値、と値が少ないので、配列を使わずにごり押しで最大値を求めてみました。配列を使う方法については今までの解答を見ればわかるはずです。その場合受け取った引数をいったん配列に入れる必要があります。

```

//整数を 3 つ受け取って、最大値を返す関数
#include<stdio.h>
int max(int a,int b,int c);
int main(void){
    int x,y,z;
    scanf("%d%d%d",&x,&y,&z); //x,y,z の値をキーボードから入力
    printf("最大の値は%d です\n",max(x,y,z)); //最大値を表示
    return 0;
}
int max(int a,int b,int c){//配列を使ってもいいかも
    int m; //最大値を保存するための変数
    //まずは a,b でどちらが大きいか. 大きいほうをとりあえず最大値とする
    if(a>=b){
        m=a;
    }
    else{
        m=b;
    } //c が m より大きいか否か
    if(m>=c){
        ; //m より小さいなら何もしない（;だけの文は空文と言います）
    }
    else{
        m=c; //c のほうが大きいなら c が最大値ということになる
    }
    return m;
}

```

3. 整数の値を 2 つ(x,y)、引数として受け取って x の y 乗を返す関数。ただし y は 0 以上の整数とします。

```
int power(int x,int y)
```

x の y 乗とは、x を y 回だけ掛けることです。このことを for 文などのループ文で実現してやればいいわけです。ちなみに四則演算子 (+, -, *, /) のような演算子は、べき乗には定義されてません。

```

//x,y の値を受け取って x の y 乗を返す関数
#include<stdio.h>
int power(int x,int y);

```

```

int main(void){
    int x,y;
    printf("x の y 乗。x と y を入力してください¥n");
    scanf("%d%d",&x,&y);
    printf("%d^%d=%d¥n",x,y,power(x,y));
    return 0;
}
int power(int x,int y){
    int k=1,i;
    for(i=0;i<y;i++){
        k=k*x;
    }
    return k;
}

```

4. 整数の値を 4 つ引数として受け取って大きい順に出力する関数

```
void order(int a,int b,int c,int d)
```

大きさを比べる関数で、引数が 4 つ以上になると問 2 のようにごり押しするのは大変です。そこで関数側で配列を用意して、引数をそれぞれ代入し、for 文を使って大きい順に並び変えています。

```

//整数を 4 つ受け取って大きい順に並び替えて表示
#include<stdio.h>
void order(int a,int b,int c,int d);
//関数プロトタイプ宣言
int main(void){
    int w,x,y,z;
    printf("整数を 4 つ入力してください¥n");
    scanf("%d%d%d%d",&w,&x,&y,&z); //整数を 4 つ入力
    printf("¥n");           //見やすくするため 1 行開ける
    order(w,x,y,z);        //関数呼び出し
    return 0;
}
void order(int a,int b,int c,int d){
    int sort[4],i,j,k;
    sort[0]=a;

```

```

sort[1]=b;
sort[2]=c;
sort[3]=d;
for(i=0;i<3;i++){ //大きい順に並び替える
    for(j=3;j>i;j--){
        if(sort[j-1]>sort[j]){
            k=sort[j];
            sort[j]=sort[j-1];
            sort[j-1]=k;
        }
    }
} //並べ替え終わったら表示
for(i=3;i>=0;i--){
    printf("%d¥n",sort[i]);
}
}

```

大きい順への並び替えですが、何をしているのかというと、配列の隣同士を右側から比べて、右側のがちいさかったら入れ替える。という操作を繰り返します。1 巡すると一番左に一番小さい数が入っていることになるので、次は一番左の数を除いて同じ操作をします。これを最後の 1 つになるまで続けています。