

[1]

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

```
main()
```

```
{
```

```
    printf("Hello, world.");
```

```
    return 0;
```

```
}
```

(G1) 使用可能な変数には以下のものがある .

int	整数	-32768 ~ 32767
-----	----	----------------

long int	整数	-2147483648 ~ 2147483647
----------	----	--------------------------

float	実数	$\pm 3.4 \times 10^{-38} \sim \pm 3.4 \times 10^{38}$
-------	----	---

double	実数	$\pm 1.7 \times 10^{-308} \sim \pm 1.7 \times 10^{308}$
--------	----	---

char	文字	
------	----	--

[2]

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

```
main()
```

```
{
```

```
    int a, b, c, d, e, f;
```

```
    printf("a = ?\n");
```

```

scanf("%d", &a);

printf("b = ?\n");
scanf("%d", &b);

c = a + b;
d = a - b;
e = a * b;
f = a / b;

printf("%d + %d = %d\n", a, b, c);
printf("%d - %d = %d\n", a, b, d);
printf("%d * %d = %d\n", a, b, e);
printf("%d / %d = %d\n", a, b, f);

return 0;
}

```

(G2) ライブラリー (既成の関数群) として, 数学の標準的関数はすべて用意されている. 数学の関数を使用するには `math.h` をインクルードし, メイク時に明示的にこの関数たちのライブラリーをリンクする必要がある.

例. `gcc -o ***** -lm *****.c`

[3]

```
#include <stdio.h>
#include <math.h>
#define PI 3.1415926
#define HALF 180.0

main()
{
    float degree, theta, s, c;

    printf("degree = ?");
    scanf("%f", &degree);

    theta = ( degree / HALF ) * PI;
    s = sin(theta);
    c = cos(theta);

    printf("cos %f = %f\n", degree, c);
    printf("sin %f = %f\n", degree, s);
    printf("cos %f ^2 + sin %f ^2 = %f\n",
           degree, degree, pow(c,2) +
           pow(s,2));
```

```
    return 0;
}
```

(G3) 制御構文 (繰り返し) : 同じ処理を複数回繰り返し行わせるには , 次の二つの構文を用いる .

for 文

```
for(i = 1; i <= n; i++){
    (処理内容)
}
```

while 文

```
while( 条件式 ){
    処理内容
}
```

For 文は繰り返し回数がそれ以前の処理で決まっている場合 , while 文は繰り返し回数が , 繰り返される処理の結果に応じて決まる場合に , それぞれ用いる .

[4]

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

```
main()
{
```

```

    int i, n, s;

    printf("n = ?\n");
    scanf("%d", &n);

    s = 0;

    for(i = 1; i <= n; i++){
        s = s + i;
    }

    printf("sum from 1 to %d = %d\n", n, s);

    return 0;
}

```

[5]

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

```

main()
{
    double aexp, term;
    int i, n;

```

```

printf("n = ?\n");
scanf("%d", &n);

aexp = 1;
term = 1;

for(i = 1; i <= n; i++){
    term = term / i;
    aexp = aexp + term;
}

printf("approximation of exp by terms
      of degree <= %d = %.15f\n", n,
      aexp);

return 0;
}

```

[6]

```

#include <stdio.h>

main()
{
    int a, b, q, r;

```

```

printf("a = ?");
scanf("%d", &a);
printf("b = ?");
scanf("%d", &b);

r = a;
q = 0;

while(r >= b){
    r = r - b;
    q = q + 1;
}

printf("%d / %d = %d *** %d \n", a, b,
        q, r);

return 0;
}

```

(G3) 制御構文 (条件分岐): 条件の成立・不成立に応じて, 処理を変更する .

```

if( 条件式 ){

    処理内容(条件式が成立した場合)
}

```

*) 不成立の場合 , この部分は実行しない

}

また ,

```
if( 条件式 ){
```

```
    処理内容(条件式が成立した場合)
```

```
}
```

```
else{
```

```
    処理内容(条件式が不成立の場合)
```

```
}
```

[7]

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

```
main()
```

```
{
```

```
    int a[5], temp, i, j;
```

```
    for(i = 0; i <= 4; i++){
```

```
        printf("a[%d]= ?", i+1);
```

```
        scanf("%d", &a[i]);
```

```
}
```

```

printf("Numbers you set; \n");
for(i = 0;i <= 4;i++){
    printf("a[%d] = %d\n", i+1, a[i]);
}

for(i = 0;i <= 3;i++){
    for(j = i+1;j <= 4;j++){
        if(a[i] < a[j]){
            temp = a[j];
            a[j] = a[i];
            a[i] = temp;
        }
    }
}

printf("Arranged numbers are\n");
for(i = 0;i <= 4;i++){
    printf("%d \t", a[i]);
}
printf("\n");

return 0;
}

```

[8]

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

```
main()
```

```
{
```

```
    float a[10][10], b[10][10], c[10][10];
```

```
    int    i, j, k, n;
```

```
    printf("dimension n=? (<=10)");
```

```
    scanf("%d", &n);
```

```
    printf("Input the entries of A\n");
```

```
    for(i = 1; i <= n; i++){
```

```
        for(j = 1; j <= n; j++){
```

```
            printf("a[%d][%d] = ?", i, j);
```

```
            scanf("%f", &a[i-1][j-1]);
```

```
        }
```

```
    }
```

```
    printf("Matrix A = \n");
```

```
    for(i = 1; i <= n; i++){
```

```
        for(j = 1; j <= n; j++){
```

```
            printf("%f \t", a[i-1][j-1]);
```

```

    }
    printf("\n");
}

printf("Input the entries of B\n");
for(i = 1;i <= n;i++){
    for(j = 1;j <= n;j++){
        printf("b[%d] [%d] = ?", i, j);
        scanf("%f", &b[i-1][j-1]);
    }
}

printf("Matrix B = \n");
for(i = 1;i <= n;i++){
    for(j = 1;j <= n;j++){
        printf("%f \t", b[i-1][j-1]);
    }
    printf("\n");
}

for(i = 1;i <= n;i++){
    for(j = 1;j <= n;j++){
        c[i-1][j-1] = 0;
        for(k = 1;k <= n;k++){

```

```

        c[i-1][j-1] = c[i-1][j-1] +
        a[i-1][k-1] * b[k-1][j-1];
    }
}
}

printf("Matrix C = A * B =\n");
for(i = 1; i <= n; i++){
    for(j = 1; j <= n; j++){
        printf("%f \t", c[i-1][j-1]);
    }
    printf("\n");
}

return 0;
}

```

(G4) 関数：関数はmain関数の前で，そのプロトタイプを宣言する．

プロトタイプ宣言：戻り値 func(引数)

例：int func(int a, float b, char c);

(G5) ポインター：他の変数のアドレスを格納し，その変数进行操作するための変数．

例：float *a, b; と宣言すると，

`a = &b`と`b`のアドレスを`a`にセットすることで, `*a`に対する操作は`b`に対する操作となる.

つまり, `*a`は任意の変数のアドレスをセットすることでその変数に対するリモコンとなる.

関数の内部から, その外側の変数を操作することはポインターを使用しない限り, できない.

[9]

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

```
int how_many(void);
```

```
void number_i(int m, int *b);
```

```
void rearr(int m, int *b);
```

```
void result_v(int m, int *b);
```

```
main()
```

```
{
```

```
    int a[10], n;
```

```
    printf("How many numbers do you input ?  
          (<= 10)\n");
```

```
    n = how_many();
```

```
    printf("Input numbers\n");
```

```

    number_i(n, &a[0]);

    rearr(n, &a[0]);

    printf("Rearranged numbers are : \n");
    result_v(n, &a[0]);

    return 0;
}

int how_many(void)
{
    int k;

    scanf("%d", &k);

    return k;
}

void number_i(int m, int *b)
{
    int i;

```

```

    for(i = 1;i <= m;i++){
        printf("number - %d - = ?", i);
        scanf("%d", b);
        if(i != m){
            b = b + 1;
        }
    }
}

```

```

void rearr(int m, int *b)
{
    int temp, i, j;

    for(i = 1;i <= m-1;i++){
        for(j = 1;j <= m-i;j++){
            if(*b < *(b+j)){
                temp = *(b+j);
                *(b+j) = *b;
                *b = temp;
            }
        }
        if(i != m-1){
            b = b + 1;
        }
    }
}

```

```

    }
}

void result_v(int m, int *b)
{
    int i;

    for(i = 1; i <= m; i++){
        printf("%d \t", *b);
        if(i != m){
            b = b + 1;
        }
    }

    printf("\n");
}

```