2023 Dec 13

HW10
說明

F74126107

吳岱昀

```
double power(double x, int n)
{
    double dbResult = 0;
    dbResult = pow(x, n);
    return dbResult;
}
```

```
double multiply(double x, int n)
{
    double dbResult = 0;
    dbResult = x * n;
    return dbResult;
}
```

```
double divide(double x, int n)
{
    double dbResult = 0;
    dbResult = x / n;
    return dbResult;
}
```

- double power(double, int) that calculates x^n if we call power(x, n) (already included <math.h>)
- double multiply(double, int) that calculates x*n if we call multiply(x, n)
- double divide(double, int) that calculates x/n if we call divide(x, n)

```
typedef double (*F)(double, int);
```

- use typedef to define a new type F which is a pointer to function

```
double powerpower(F ppfCal, double x, int n, int m)
{
    double dbResult = 0;

    dbResult = ppfCal(x,n);
    dbResult = pow(dbResult,m);

    return dbResult;
}
```

- declare ppfCal as a pointer to function (type is F)

- then can caculate the result of (x^n)^m, (x*n)^m, (x/n)^m

```c
int main(int argc, char *argv[]) {
    F pf_Cal;
    double dbl_x;
    int n;
    int m;
    double dbl_result;
```

- use argc, argv to get the parameters in the comand line

```c
int i;
printf("argc:%d\n", argc);
for( i = 0; i < argc; i += 1)
{
    printf("%d. %s\n", i, argv[i]);
}
printf("\n");
```

- print and check the parameter number

```c
if( argc < 4 )
{
    printf("[Error] Invalid parameter number! argc:%d\n", argc);
    return -1;
}
```

```c
// Get para-1 ...
sscanf(argv[1], "%lf", &dbl_x);

// Get para-2 ...
sscanf(argv[2], "%d", &n);

// Get para-3 ...
sscanf(argv[3], "%d", &m);

printf("Para: x:%lf, n:%d, m:%d\n", dbl_x, n, m);
```

- use sscanf to get the parameters (x, n, m)

```c
dbl_result = powerpower(power, dbl_x, n, m);
printf("powerpower cal: power:  x:%lf, n:%d, m:%d => %lf\n\n", dbl_x, n, m, dbl_result);

dbl_result = powerpower(multiply, dbl_x, n, m);
printf("powerpower cal: multiply:  x:%lf, n:%d, m:%d => %lf\n\n", dbl_x, n, m, dbl_result);

dbl_result = powerpower(divide, dbl_x, n, m);
printf("powerpower cal: divide:  x:%lf, n:%d, m:%d => %lf\n\n", dbl_x, n, m, dbl_result);
```

- call powerpower by using parameters

Thanks for watcning.