

Group Work: choose the ones you like and do in group of 3-5 people.

1. Write a function that returns:

a)  $n!$

Let me know if you can't write this one within 1 minute!

b)  $1/1^k + 1/2^k + \dots + 1/n^k$  where  $k$  is positive integer, you should not use math library

NOTE: the question ask you to write a function, so just write a function. Don't write `main()`, and (normally) you should not use `scanf` or even `printf`.

You need to write a loop or a function to compute  $n^k$ , you cannot use  $n**k$  because there is no "power operator" in C. Also, be careful:  $1/s$  is zero if  $s$  is an integer and is greater than 1.

It is ok if you want to write a recursive function for this question, because a recursive function is *simpler*.

c)  $1 - x + x^2/2! - x^3/3! + \dots + (-1)^n x^n/n!$  where  $n$  is the smallest positive integer that satisfies  $|x^n/n!| < 10^{-6}$ . You should not write a separate function for  $x^n$  or  $n!$

Have a look at the members of the above sum. You can see that two consecutive members are quite similar. And In fact if we know the value of one member, we can easily compute the value of the next one, just by multiply the former by  $x$ , and then divide the result by a single value.

Quick solution:

```
1 #define EPS 1e-6
2
3 double sum(double x) {
4     int i=0;
5     double p=1, s=1;
6     // when i=0, s and p already include respective values!
7
8     while (1) { // we can make infinit loop and exit when needed
9         i++; // at first iteration, i becomes 1
10        // NOW s and p are "behind" i
11        p = -p*x/i; // first iteration: check that p= -x/1!
12        s += p; // now, s and p all updated with i
13        if (fabs(p) < EPS) { // check: is the last iteration correct?
14            break;
15        }
16    }
17    return s;
18 }
```

2. Write function

`int f( int n, int k, double *psum)`

that computes the sum 1b) above. If  $n < 1$  or  $k < 0$ , the function should return `INVALID` and should not modify the variable indicated by pointer `psum`. Otherwise the function should return `VALID`, and the variable indicated by `psum` should have the value of the sum.

Note: The solution is clear now after you read sample test 1.

3. Write a program that draw a triangle of high n. For example, when n=3 the triangle should be:

```
  *
 * *
* * *
```

Solution: Note that there is a space between 2 consecutive stars. And so, in each line we need to print a number of " " (a string containing exactly two spaces) and a number of "\*" – a string containing one star and one space. Also, we need to print n lines, starting with a line with has (n-1) double spaces and 1 star, and finished with the line that has no spaces at all.

Note also that this question asks to write a program, which means we need to write a main() function, and also need to perform input/output with scanf/printf.

Program:

```
1 #include <stdio.h>
2
3 // the task seems to be ok only when n is positive and is not too big.
4 // we define MIN and MAX as min and max for n
5 #define MIN 1
6 #define MAX 20
7
8 int main(int argc, char *argv[]) {
9     int n, i, j;
10
11     printf("Enter value for n:");
12     while ( scanf("%d", &n) == 1 ) {
13         if (n>=MIN && n<=MAX) {
14             break;
15         }
16         printf("Enter value between %d and %d:", MIN, MAX);
17     }
18
19     for (i=0; i<n; i++) { // print n lines
20         int stars= i+1; // number of stars, which is 1 when i=0
21         int spaces= n-i; // stars and spaces add up to n
22         for (j=0; j<spaces; j++) {
23             printf(" ");
24         }
25         for (j=0; j<stars; j++) {
26             printf(" *");
27         }
28         printf("\n"); // end line
29     }
30     return 0;
31 }
```

3. As described in exercises 7.8, 7.9, 7.10. That is, supposing A is an array of n integers, write a function that:

- 7.8: returns the k-th smallest of A, not modifying A
- 7.9: returns the number of runs in A (a *run* is a sequence of non-increasing values)
- 7.10: returns the number of inversions in A (an *inversion* is a pair (a, b) where a>b)

Solution: - for these questions, please see workshop solutions that will be provided in the LMS at some stage.