C Programming
Lab 4: Loops

Lecturer: *Dr*. Wan-Lei Zhao

Spring Semester 2022

Email: wlzhao@xmu.edu.cn,

Approximate exp(x) (1)

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1} + \frac{x^2}{2} + \frac{x^3}{6} \ldots \quad (1)$$

- Requirements
  - Keep terms that larger than 1e-6
- Hints
  - Define term=$\frac{x^n}{n!}$
  - Do loop while abs(term) is larger than 1e-6

Approximate exp(x) (2)

```c
#include <stdio.h>
#include <math.h>
int main()
{
   const double prec = 1e-6;
   double term = 1, sum = 0, x =0.3;
   double up = 1, low = 1;
   int i = 1;
   scanf("%lf", &x);
   while(abs(term) > prec)
   {
      sum += term;
       up  = up*x;
       low = low*i;
      term = up/low;
      i++;
   }
   printf("%lf\n", sum);
}
```

Print pyramid of alphabets on the screen

A
BBB
CCCCC
DDDDDDD
EEEEEEEEE

- Hints
  - Suggested to use **for** loop
  - Two levels of embeding

```
1 #include <stdio.h>
2 for(i = 0; i < 5; i++)}
3 {
4     for(j = 0; j < ?; j++)}
5     {
6       //filling your code
7     }
8 }
```

Answer

```
#include <stdio.h>
int main()
{
   int i=0, j=0, count=0;
   int nl=5, nc=1, nb=nl-1;
   for(j = 0; j < nl; j++)
   {
      for(i = 0; i < nb; i++)
      {
         printf(" ");
      }
      nc = 2*j-1;
      for(i=0;i<nc;i++)
      {
         printf("%c", ch);
      }
      ch++;
      nb--;
      printf("\n");
   }//for(j)
}
```

Find out prime numbers

- Requirements
  - Find all prime numbers smaller than **500**
  - Print out 8 numbers on each line
- Hints
  - Only dividable by itself
  - By filtering method
  - Try "%" from **2** to **sqrt(num)**

Answer (1)

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      int _PRIME_ = 1;
7      float b = 0;
8      int i = 0, j = 0, count = 0;
9      for(i = 2; i <= 500; i++)
10     {
11         b = sqrt(i+0.0);
12         _PRIME_ = 1;
13         for(j = 2; j < b; j++)
14         {
15             if(i%j == 0 && i != 2)
16             {
17                 _PRIME_ = 0;
```

Answer (2)

```
18              }//if(i%j)
19          }//for(j)
20          if(_PRIME_ == 1)
21          {
22              count++;
23              printf("%d\t", i);
24              if(count%8 == 0)
25              {
26                  printf("\n");
27              }
28          }//if
29      }//for(i)
30      if(count %8 !=0)
31          printf("\n");
32  }
```

Find out all complete numbers

- Find out all the **complete number** in the range of [1, 10000]
- Complete number: it equals to the sum of its factors
- Example: $6 = 1+2+3$

Answer

```c
#include <stdio.h>
int main()
{
  int j=0, i=0, sum=0;
  for(j=1; j<=10000; j++)
  {
     sum = 1;
     for(i = 2; i < j; i++)
     {
        if(j%i == 0)
        {
           sum += i;
        }
     }//for(i)
     if(sum == j)
     {
        printf("%d\t", j);
     }
   }//for(j)
   printf("\n");
}
```

Convert pure decimal fraction into binary form

- Convert pure decimal fraction such as '0.635' into its binary form '0.10100010100011110101111'
- Accept a pure decimal fraction from input: 0.625
- Output its binary form: 0.101
- The loop continues until the fraction is lower than 0.005

Answer

```c
#include <stdio.h>
int main()
{
    float a = 0.635;
    scanf("%f", &a);
    if(a < 1.0){
      printf("0.");
      do{
          a = a*2;
          if(a >= 1.0)
          {
              printf("1");
              a = a - 1.0;
          }else{
             printf("0");
          }
      }while(a > 0.005);
    }//end if
    return 0;
}
```