CSE102#hw04

Due Date: March 25, 23:55

In this homework you will write programs to exercise with loops, functions and pointers. Please do not forget that your program should work exactly as the example runs given below the part definitions. Submit your homework via KADI.

#part1

In this part of the homework, you will use Taylor series to calculate sine and cosine.

$$sinx = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots for\ all\ x$$

$$cosx = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots for\ all\ x$$

The user will enter an integer as degree and another integer that defines the first n elements of the series. **You are not allowed to use math library functions** for calculations, that is you must write your own functions ***with the same input/output format*** of the math library function. In addition, use the following function signatures to make your program modular:

```
double cosine(int degree, int n);
double sine(int degree, int n);
void getInputs(int* degree, int* n, int* exit);
```

The function named ***getInputs*** will wait for user to enter the ***degree*** and ***n*** value. If **'E'** is entered instead, the ***exit*** variable will be 1, otherwise 0.

#sample run

```
30 4
sin(30) where n is 4 = 0.4998
cos(30) where n is 4 = 0.8662
60 2
sin(60) where n is 2 = 0.8556
cos(60) where n is 2 = 0.4522
45 3
sin(45) where n is 3 = 0.7069
cos(45) where n is 3 = 0.7077
e
```

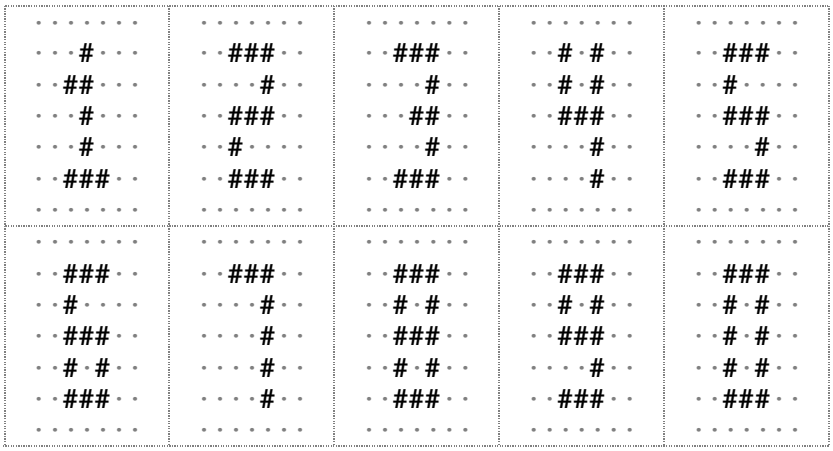
#part2

In this part of the homework you will get a positive integer from the user and print out a simple ASCII art. An example run of the program is given below. The program will not print out any message and wait input from the user. Beside the number to be printed, the user will enter the width of the border and the character to be used while printing.

Regardless from the input data, each digit will be printed in an area of 7x7 (rows and columns) as shown below. User should enter at most 5-digit integer, your program should print nothing if integer is higher than 99999. The program will wait for the input from the user until he enters a negative number. Use the getInputs() function described below to read input from the user.

```
void getInputs(int* num, int* borderWidth, char* borderChar);
int printNumber(int num, int borderWidth, char borderChar);
```

Both functions' parameters are the number to be printed, the border width and the character to be used while printing. Number formats are defined below, you must use them as defined. printNumber function returns 0 if number is invalid, otherwise 1.



#sample run:

User input: 693 Border width: 1 The character is '#'

```
693 1 #¬
########################¬
#····················#¬
#··###····###····###··#¬
#··#·····#·#·····#··#¬
#··###····###·····##··#¬
#··#·#·····#·····#··#¬
#··###····###····###··#¬
#····················#¬
########################¬
```

User input: 693 Border width: 2 The character is '8'

```
693 2 8¬
8888888888888888888888888¬
8888888888888888888888888¬
88····················88¬
88··888····888····888··88¬
88··8·····8·8·····8··88¬
88··888····888·····88··88¬
88··8·8·····8·····8··88¬
88··888····888····888··88¬
88····················88¬
8888888888888888888888888¬
8888888888888888888888888¬
```

Note that spaces are shown as '·'.

NOTES:

- **You should not send your main function. You should send the functions described above and your helper functions if you have any.**
- Use the structures you learned such as loops, conditions and functions whenever needed. The usage of these structures is as important as the correct execution of your program.
- The assignment must be your original work. Duplicate or very similar assignments are both going to be considered as cheating.
- Ask your questions via moodle.