**PATUAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY**

**COURSE CODE CIT-112**

**SUBMITTED TO:**

### **Md. Mahbubur Rahman**

### **Department of Computer Science and Information Technology**

**Faculty of Computer Science and Engineering**

**SUBMITTED BY:**

**Md. Sharafat Karim**

ID: **2102024**,

Registration No: **10151**

**Faculty of Computer Science and Engineering**

Date of submission: **30 May, 2023**

**Assignment: 06 (Basic Code)**

Table of Contents

[1) Reversing digits 3](#__RefHeading___Toc2159_601131689)

[2) Euler’s Number 4](#__RefHeading___Toc2184_601131689)

[3) Functions calculate 5](#__RefHeading___Toc2186_601131689)

[4) Present value 7](#__RefHeading___Toc2188_601131689)

[5) Square of character S 9](#__RefHeading___Toc2190_601131689)

[6) sin(x) graph function 11](#__RefHeading___Toc2192_601131689)

[7) S square with middle O 13](#__RefHeading___Toc2194_601131689)

[8) postivie values 14](#__RefHeading___Toc2196_601131689)

[9) Colored line 15](#__RefHeading___Toc2198_601131689)

[10) Pattern pyramid center aligned 16](#__RefHeading___Toc2200_601131689)

[11) Pattern of floating island 18](#__RefHeading___Toc2202_601131689)

[12) Sum of the digits 20](#__RefHeading___Toc2204_601131689)

[13) left sided pattern 21](#__RefHeading___Toc2206_601131689)

[14) Fibonacci numbers 22](#__RefHeading___Toc2208_601131689)

[15) Investment equation 23](#__RefHeading___Toc2210_601131689)

[16) pattern with for loop 25](#__RefHeading___Toc2212_601131689)

[17) Age range 27](#__RefHeading___Toc2214_601131689)

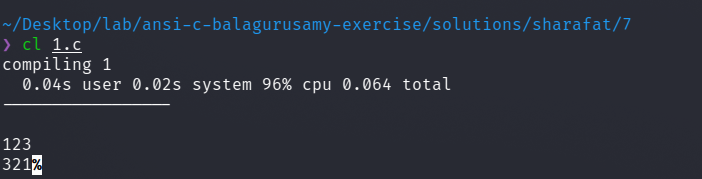
[18) 2 curves 29](#__RefHeading___Toc2216_601131689)

[19) exponent table 32](#__RefHeading___Toc2218_601131689)

[20) S writing with loop 34](#__RefHeading___Toc2220_601131689)

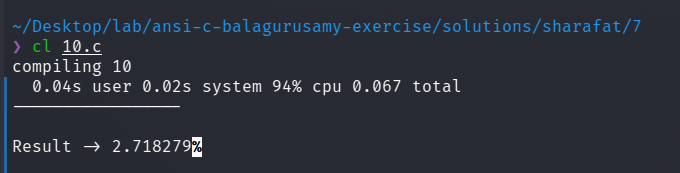
# 1) Reversing digits

#include<stdio.h>  
int main()  
{  
 int n;  
 scanf("%d", &n);  
  
 int reverse=0;  
 while (n>0)  
 {  
 reverse \*= 10;  
 reverse += n % 10;  
 n /= 10;  
 }  
 printf("%d", reverse);  
}



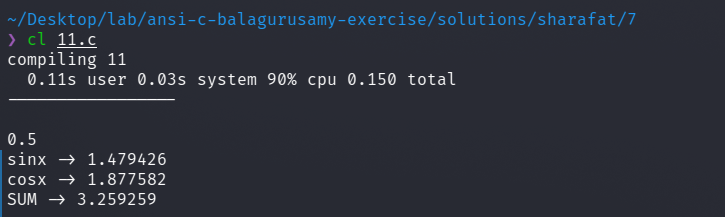
# 2) Euler’s Number

#include <stdio.h>  
  
int main()  
{  
 float accuracy = 0.00001;  
 float result=1, sum = 1;  
 int i = 1;  
  
 while ( sum > accuracy )  
 {  
 result += sum;  
 i++;  
 sum \*= ((float)1/i);  
 }  
  
 printf("Result -> %f", result);  
}



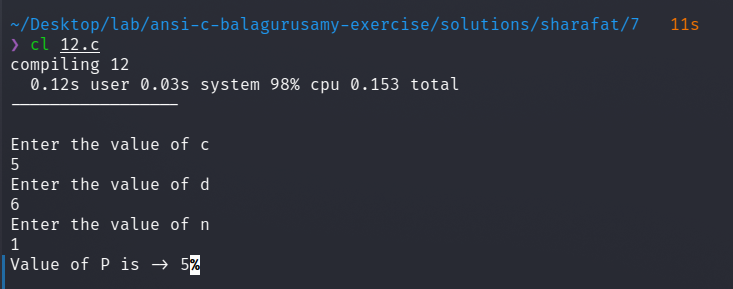
# 3) Functions calculate

#include <stdio.h>  
#include <math.h>  
  
int main()  
{  
 float accuracy = 0.0001, x;  
 scanf("%f", &x);  
 float result=1, sum = x;  
 int i = 1;  
  
 while ( sum > accuracy || sum < 0 )  
 {  
 result += sum;  
 i += 2;  
 sum \*= (-1)\*(x\*x)/(i\*(i-1));  
  
 if (i >= 100)  
 break;  
 }  
  
 printf("sinx -> %f\n", result);  
  
 result=1, sum = 1;  
 i = 0;  
  
 while ( sum > accuracy || sum < 0 )  
 {  
 result += sum;  
 i += 2;  
 sum \*= (-1)\*(x\*x)/(i\*(i-1));  
  
 if (i >= 100)  
 break;  
 }  
  
 printf("cosx -> %f\n", result);  
  
 result=1, sum = 1;  
 i = 0;  
  
 while ( sum > accuracy || sum < 0 )  
 {  
 result += sum;  
 i += 1;  
 sum \*= pow(((float)1/i),i);  
  
 if (i >= 100)  
 break;  
 }  
  
 printf("SUM -> %f\n", result);  
}



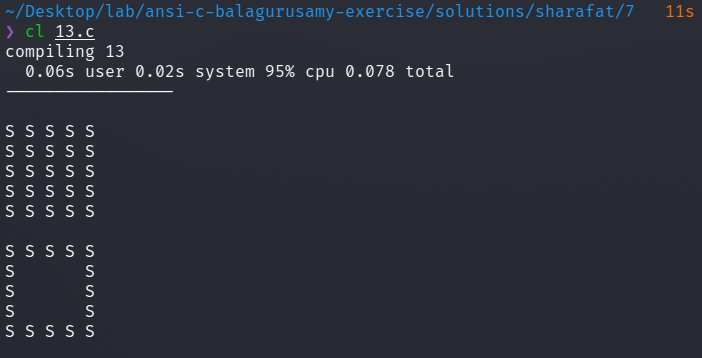
# 4) Present value

#include <math.h>  
#include <stdio.h>  
#include <ctype.h>  
  
int input(char name)  
{  
 int n;  
 printf("Enter the value of %c\n", name );  
 scanf("%d", &n);  
 if (n >= 0)  
 return n;  
 else  
 {  
 printf("Invalid input! Please try again\n");  
 printf("Value of P is -> %d", n);  
 return input(name);  
 }  
}  
  
int main()  
{  
 int n, c, d;  
 c = input('c');  
 d = input('d');  
 n = input('n');  
  
 int P;  
 P = c \* pow(( 1 - (int)(d/100) ), n);  
 printf("Value of P is -> %d", P);  
  
 return 0;  
}



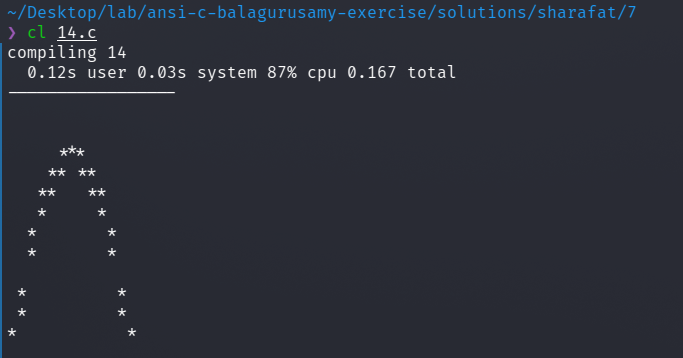
# 5) Square of character S

#include<stdio.h>  
  
int main()  
{  
 int i, j;  
 int col = 5;  
 char Char = 'S';  
  
 for (i=0; i< col; i++)  
 {  
 for (j=0; j<col; j++)  
 {  
 printf("%c ", Char);  
 }  
 printf("\n");  
 }  
  
 printf("\n");  
  
 for (i=0; i< col; i++)  
 {  
 for (j=0; j<col; j++)  
 {  
 if (i==0||j==0||i==col-1||j==col-1)  
 printf("%c ", Char);  
 else  
 printf(" ");  
 }  
 printf("\n");  
 }  
}



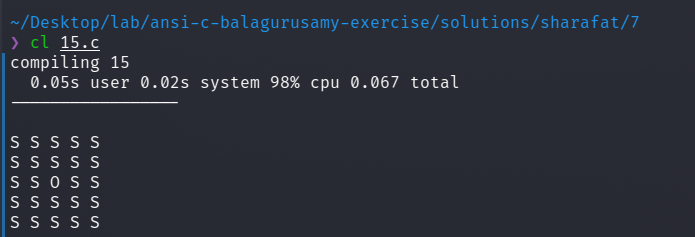
# 6) sin(x) graph function

#include <stdio.h>  
#include <math.h>  
  
/\*  
x y  
--------------------------  
0 0.000000  
15 0.258819  
30 0.500000  
45 0.707107  
60 0.866025  
75 0.965926  
90 1.000000  
105 0.965926  
120 0.866025  
135 0.707107  
150 0.500000  
165 0.258819  
180 0.000000  
\*/  
  
#define PI 3.14159265  
int main()  
{  
 int i;  
 float j, s;  
 for (j=1.1; j >= 0; j-=0.1)  
 {  
 for (i=0; i<= 180; i+= 15)  
 {  
 s = sin(i\*PI/180);  
 if (fabsf(s-j)<=0.1)  
 {  
 printf("\*");  
 }  
 else {  
 printf(" ");  
 }  
 }  
 printf("\n");  
 }  
}



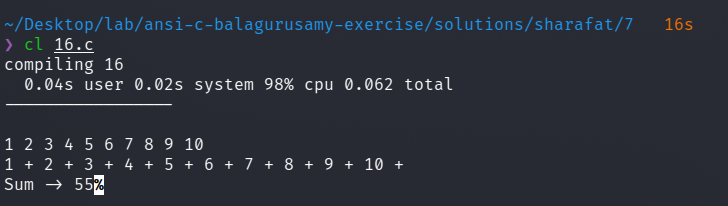
# 7) S square with middle O

#include<stdio.h>  
  
int main()  
{  
 int i, j;  
 int col = 5;  
 char Char = 'S';  
 char Char2 = 'O';  
  
 for (i=0; i< col; i++)  
 {  
 for (j=0; j<col; j++)  
 {  
 if (i==j && i==2)  
 printf("%c ", Char2);  
 else  
 printf("%c ", Char);  
 }  
 printf("\n");  
 }  
}



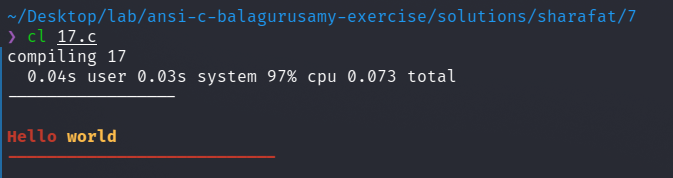
# 8) postivie values

#include<stdio.h>  
  
int main()  
{  
 int i, j;  
 int sum=0;  
  
 for (i=0; i< 10; i++)  
 {  
 scanf("%d", &j);  
 sum += j;  
 printf("%d + ", j);  
 }  
 printf("\nSum -> %d", sum);  
}  
  
// test case  
// 1 2 3 4 5 6 7 8 9 10



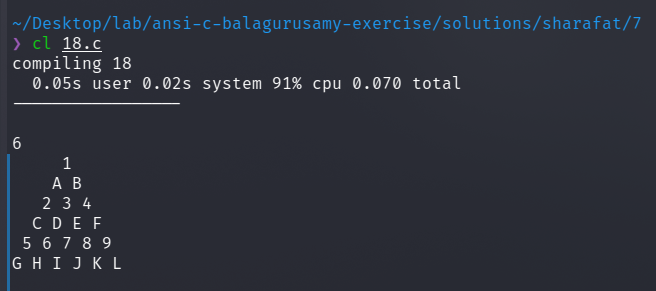
# 9) Colored line

#include <stdio.h>  
void red () {  
 printf("\033[1;31m");  
}  
  
void yellow() {  
 printf("\033[1;33m");  
}  
  
void reset () {  
 printf("\033[0m");  
}  
  
int main () {  
 red();  
 printf("Hello ");  
 yellow();  
 printf("world\n");  
 red();  
 printf("---------------------------\n");  
 reset();  
 return 0;  
}



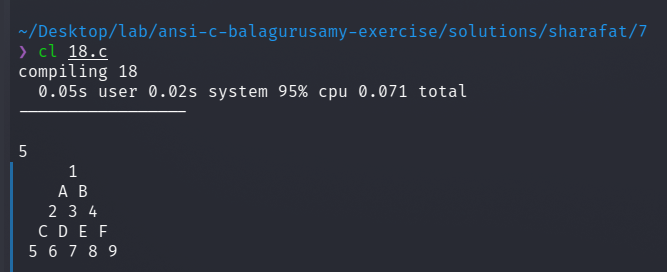
# 10) Pattern pyramid center aligned

#include<stdio.h>  
  
int main()  
{  
 int i, j;  
 int digit=1;  
 char letter='A';  
 int n;  
 scanf("%d", &n);  
  
 for (i=0; i< n; i++)  
 {  
 for (j=0; j<(5-i);j++)  
 printf(" ");  
 for (j=0; j<= i; j++)  
 {  
 if (i%2==0)  
 printf("%d ", digit++);  
 else  
 printf("%c ", letter++);  
 }  
 printf("\n");  
 }  
}



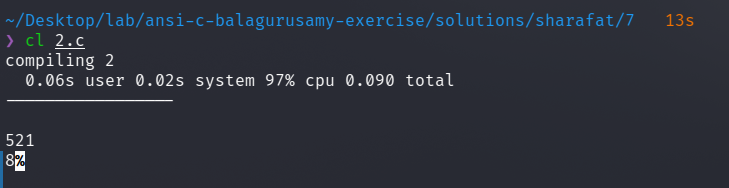
# 11) Pattern of floating island

#include<stdio.h>  
  
int main()  
{  
 int i, j;  
 int digit=1;  
 char letter='A';  
 int n;  
 scanf("%d", &n);  
  
 for (i=0; i< n/2; i++)  
 {  
 for (j=0; j<(5-i);j++)  
 printf(" ");  
 for (j=1; j<= i+1; j++)  
 printf("%d ", j);  
 for (j=j-2;j>=1;j--)  
 printf("%d ", j);  
 printf("\n");  
 }  
  
 for (i=n/2; i>= 0; i--)  
 {  
 for (j=0; j<(5-i);j++)  
 printf(" ");  
 for (j=1; j<= i+1; j++)  
 printf("%d ", j);  
 for (j=j-2;j>=1;j--)  
 printf("%d ", j);  
 printf("\n");  
 }  
}



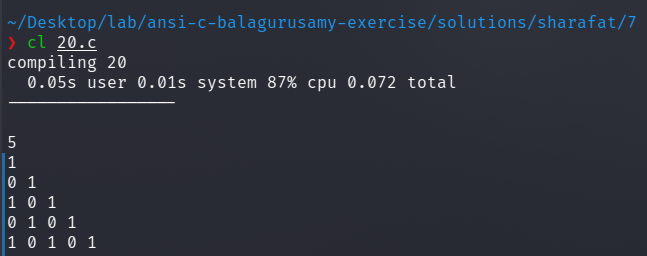
# 12) Sum of the digits

#include<stdio.h>  
int main()  
{  
 int n;  
 scanf("%d", &n);  
  
 int sum=0;  
 while (n>0)  
 {  
 sum += n % 10;  
 n /= 10;  
 }  
 printf("%d", sum);  
}



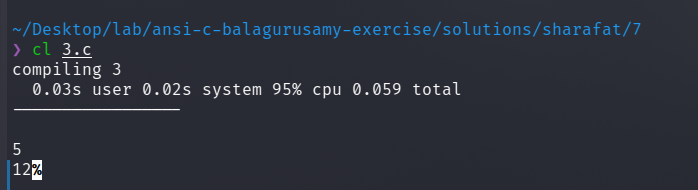
# 13) left sided pattern

#include<stdio.h>  
  
int main()  
{  
 int i, j;  
 int digit=1;  
 char letter='A';  
 int n;  
 scanf("%d", &n);  
  
 for (i=0; i< n; i++)  
 {  
 for (j=0; j<= i; j++)  
 printf("%d ", ((i+j)%2==0?1:0));  
 printf("\n");  
 }  
}



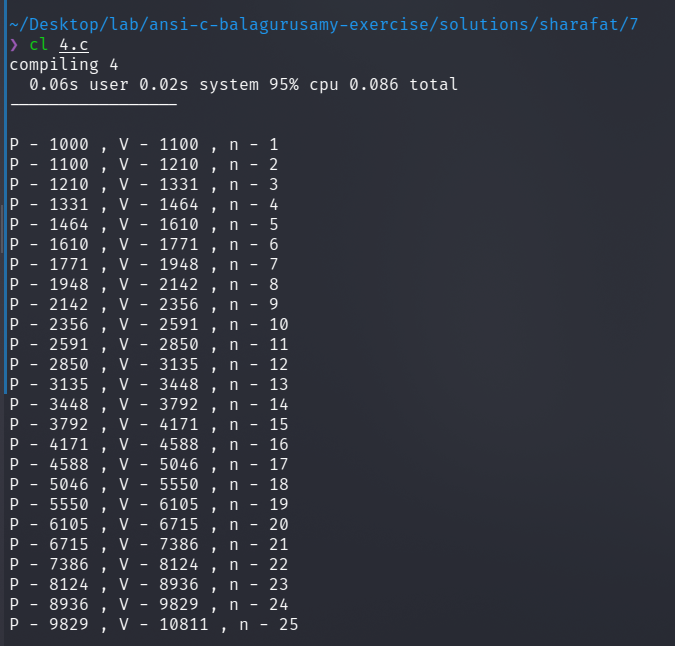
# 14) Fibonacci numbers

#include<stdio.h>  
int main()  
{  
 int n;  
 scanf("%d", &n);  
  
 int i=0, j=1, k=1, l=0, sum=0;  
 do  
 {  
 if ( i == 0 || i == 1 )  
 sum += 1;  
 else  
 {  
 sum += j + k;  
 l = j + k;  
 k = j;  
 j = l;  
 }  
 i++;  
 } while (i<n);  
 printf("%d", sum);  
}



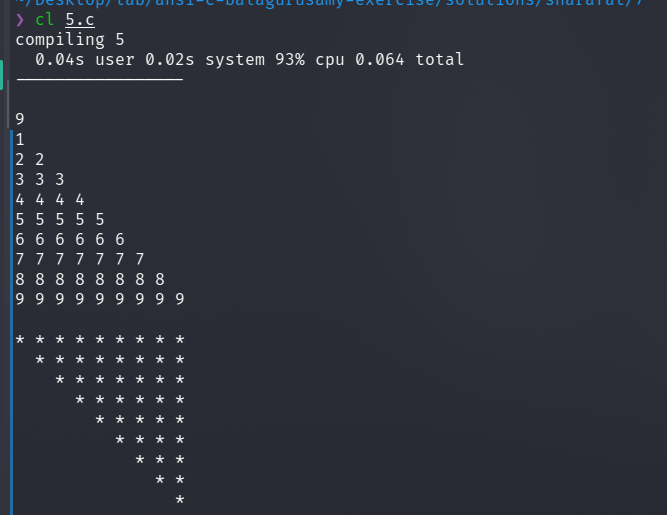
# 15) Investment equation

#include<stdio.h>  
int main()  
{  
 int n=1;  
 int p=1000, V ; float r=0.10;  
 do  
 {  
 V = p \* (1+r);  
 printf("P - %d , V - %d , n - %d \n", p, V, n++);  
 p = V;  
 } while ( p <= 10000 );  
}



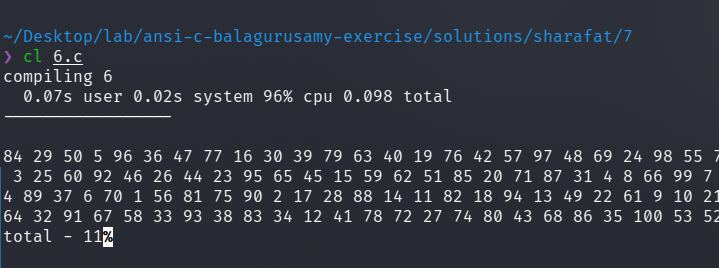
# 16) pattern with for loop

#include <stdio.h>  
  
int main()  
{  
 int n;  
 scanf("%d", &n);  
  
 int i, j;  
 for (i=0; i<n; i++)  
 {  
 for (j=1; j<=i+1; j++)  
 printf("%d ", i+1);  
 printf("\n");  
 }  
  
 printf("\n");  
  
 for (i=0; i<n; i++)  
 {  
 for (j=0; j<i; j++)  
 printf(" ");  
 for ( j=n-i ; j>0 ; j-- )  
 printf("\* ");  
 printf("\n");  
 }  
}



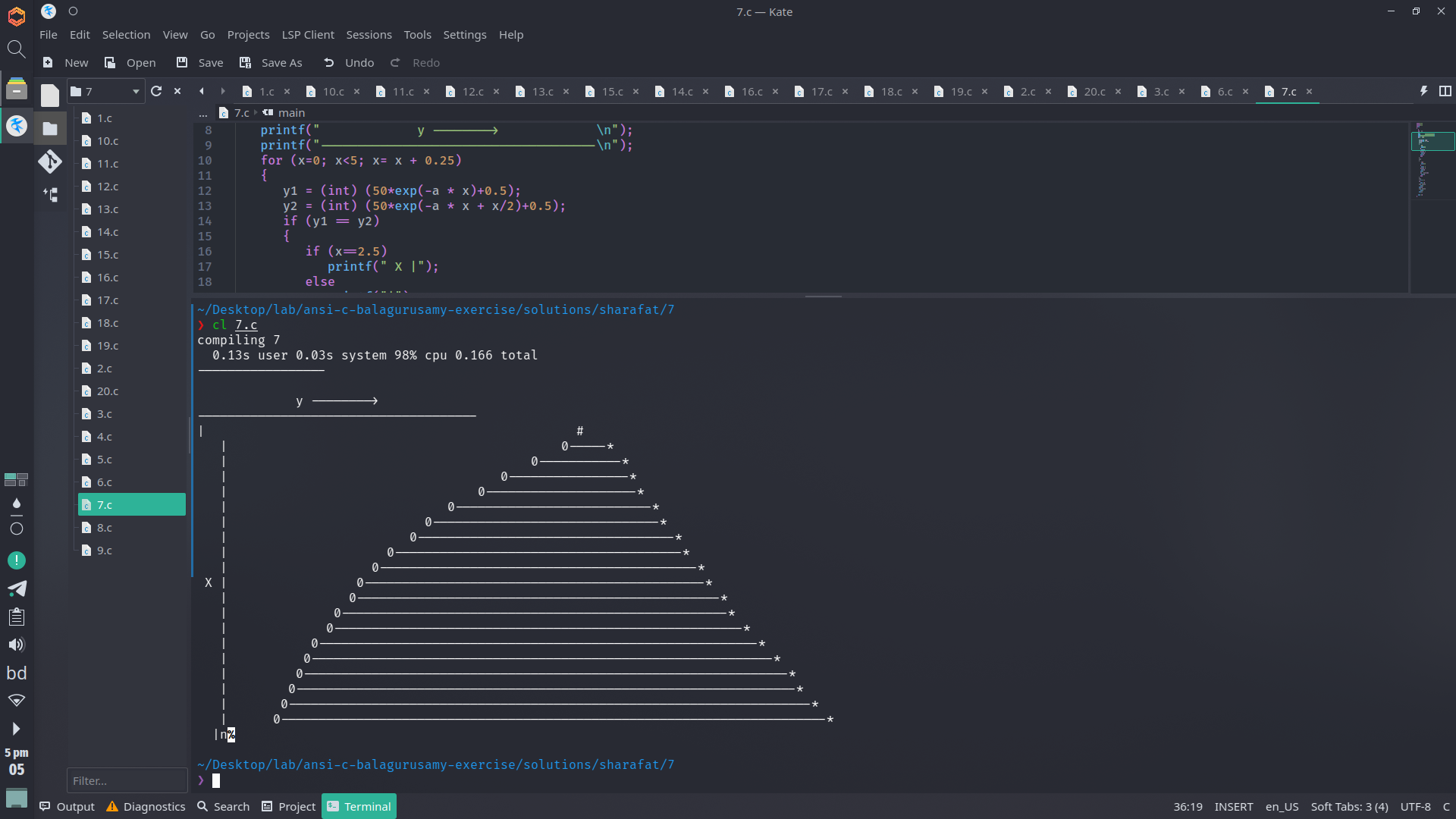
# 17) Age range

#include<stdio.h>  
  
int main()  
{  
 int i, j;  
 int sum = 0;  
  
 for (i=0; i<100; i++)  
 {  
 scanf("%d", &j);  
 if ( 50 <= j && j <= 60 )  
 {  
 sum++;  
 continue;  
 }  
 }  
 printf("total - %d", sum);  
}  
  
// test cases  
// 84 29 50 5 96 36 47 77 16 30 39 79 63 40 19 76 42 57 97 48 69 24 98 55 73 3 25 60 92 46 26 44 23 95 65 45 15 59 62 51 85 20 71 87 31 4 8 66 99 7 54 89 37 6 70 1 56 81 75 90 2 17 28 88 14 11 82 18 94 13 49 22 61 9 10 21 64 32 91 67 58 33 93 38 83 34 12 41 78 72 27 74 80 43 68 86 35 100 53 52



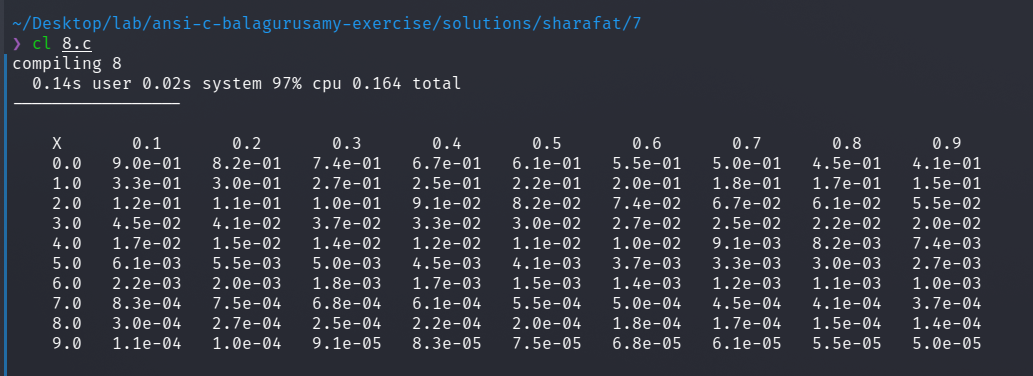
# 18) 2 curves

#include<stdio.h>  
#include <math.h>  
int main()  
{  
 int i;  
 float a, x, y1, y2;  
 a = 0.4;  
 printf(" y --------> \n");  
 printf("-------------------------------------\n");  
 for (x=0; x<5; x= x + 0.25)  
 {  
 y1 = (int) (50\*exp(-a \* x)+0.5);  
 y2 = (int) (50\*exp(-a \* x + x/2)+0.5);  
 if (y1 == y2)  
 {  
 if (x==2.5)  
 printf(" X |");  
 else  
 printf("|");  
 for (i=1; i <= y1-1; ++i)  
 printf(" ");  
 printf("#\n");  
 continue;  
 }  
 if (y1>y2)  
 {  
 if ( x == 2.5 )  
 printf(" X |");  
 else  
 printf(" |");  
 for (i=1; i <= y2-1; ++i )  
 printf(" ");  
 printf("\*");  
 for ( i = 1; i <= (y1 - y2-1);++i)  
 printf("-");  
 continue;  
 }  
 if (x==2.5)  
 printf(" X |");  
 else  
 printf(" |");  
 for (i = 1; i <= (y1 - 1); ++i)  
 printf(" ");  
 printf("0");  
 for (i = 1; i <= (y2-y1-1); ++i)  
 printf("-");  
 printf("\*\n");  
 }  
 printf(" |n");  
  
  
 float j, s;  
 for (j=1.1; j >= 0; j-=0.1)  
 {  
 for (i=0; i<= 5; i+= 0.5)  
 {  
 s = exp(-a \* x);  
 if (fabsf(s-j)<=0.1)  
 {  
 printf("\*");  
 }  
 else {  
 printf(" ");  
 }  
 }  
 printf("\n");  
 }  
}



# 19) exponent table

#include<stdio.h>  
#include <math.h>  
  
int main()  
{  
 float x, y;  
 x = 0.0;  
  
 float i, j;  
 for (i=0; i<= (float)(0.91) ; i += 0.1)  
 {  
 if (i==0)  
 printf(" X ");  
 else  
 printf(" %.1f ", i);  
 }  
 printf("\n");  
  
 for (i=0; i<= 9 ; i += 1)  
 {  
 for (j=i; j<=i+0.91; j += 0.1)  
 {  
 if (j==i)  
 printf(" %.1f ", j);  
 else  
 printf(" %.1e ", exp(-j));  
 }  
 printf("\n");  
 }  
 printf("\n");  
}



# 20) S writing with loop

#include <stdio.h>  
  
int main() {  
 int rows = 15;  
 int cols = 18;  
 int i, j;  
  
 for (i = 0; i < cols; i++)  
 printf("\*");  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 2 || i > 15)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 9 || i > 15)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 4 )  
 printf("\*");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 4 )  
 printf("\*");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 4 )  
 printf("\*");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 5 || i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
 for (i = 0; i < cols; i++)  
 {  
 if (i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
 for (i = 0; i < cols; i++)  
 {  
 if (i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
 for (i = 0; i < cols; i++)  
 {  
 if (i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
 for (i = 0; i < cols; i++)  
 {  
 if (i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 4 || i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 3 || i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 for (i = 0; i < cols; i++)  
 {  
 if (i < 2 || i > 13)  
 printf("\*");  
 else  
 printf("-");  
 }  
 printf("\n");  
  
 return 0;  
}

