**ADITYA AGARWAL 21104050**

**VIPUL VERMA 21104052**

**ARIN JAIN 21104049**

**ATHARV VIJAYVERGIA 21104048**

**SDF-2 MINI PROJECT**

**PRODUCTIVITY INTERFACE**

Our productivity interface consists of 4 parts:

\*Reflex game

\*Timer

\*Calculator

\*Journal (To do list)

So now our code uses the concepts of OOP i.e., object orientation programming, concepts of vector, STL, inheritance, sleep function, functions, switch cases (control statements) and much more. But the main and important thing used in code is **multi-threading** which allows the compiler to take input from the user while simultaneously executing the code which opens more possibilities for the code.

The interface starts with 4 options with a square frame used to select the option named as {R, T, C, J}. The frame is made with the help of 2-dimension arrays. It is done to give used a better experience for selecting the option.

Next is the reflex game in which big alphabetical letters will be displayed to the user and he/she must type that character in that fraction of time to get the score achieved. Reflex game was made with big characters with random function (rand function). Big characters are used with frame usage and in the end the score is displayed to the user.

Next is the **Timer** which asks the user to give the hours, minutes, and seconds and gives option to user by giving in built defined timer for saving users time. Also, the function has feature to add 1 minute while timer going on and even reduce by 1 minute using multi-thread function.

After **timer** next is **calculator** which takes the operation, and the numbers from the user and give the resultant to the user which saves time of the user in calculation stuff hence improving the productivity of the user.

Since we are done with the **calculator, next** part is **journal** which is TO DO LIST which helps the user to write, edit and add task to the file in a document which will also increase productivity of the user. Now the journal code uses the concept of File Handling for reading, writing, and editing the to do list document of the user.

All this was the basic information of our mini-SDF 2 project, we will hope that you will like it.

**proj.cpp**

#include <iostream>

#include <windows.h>

#include <chrono>

#include <thread>

#include <conio.h>

//#include <time.h>

//#include <bits/stdc++.h>

#include "timerup.cpp"

#include "sclc.cpp"

#include "reflex.cpp"

#include "td.cpp"

using namespace std;

/\*using std::chrono::duration\_cast;

using std::chrono::milliseconds;

using std::chrono::seconds;

using std::chrono::system\_clock;\*/

char frame[50][100];

class selection\_screen

{

public:

void first\_screen(int selection)

{

for (int i = 0; i < 50; i++)

{

for (int j = 0; j < 100; j++)

frame[i][j] = ' ';

frame[11][22] = '@';

frame[11][23] = '@';

frame[11][24] = '@';

frame[11][25] = '@';

frame[11][26] = '@';

frame[12][22] = '@';

frame[13][22] = '@';

frame[14][22] = '@';

frame[15][22] = '@';

frame[16][22] = '@';

frame[17][22] = '@';

frame[12][27] = '@';

frame[13][27] = '@';

frame[15][24] = '@';

frame[14][23] = '@';

frame[14][24] = '@';

frame[14][25] = '@';

frame[16][25] = '@';

frame[17][26] = '@';

frame[31][22] = '@';

frame[31][23] = '@';

frame[31][24] = '@';

frame[31][25] = '@';

frame[31][26] = '@';

frame[31][27] = '@';

frame[31][28] = '@';

frame[32][26] = '@';

frame[33][26] = '@';

frame[34][26] = '@';

frame[35][26] = '@';

frame[36][26] = '@';

frame[37][25] = '@';

frame[37][24] = '@';

frame[36][23] = '@';

frame[31][61] = '@';

frame[31][62] = '@';

frame[31][63] = '@';

frame[31][64] = '@';

frame[31][65] = '@';

frame[31][66] = '@';

frame[31][67] = '@';

frame[31][68] = '@';

frame[31][69] = '@';

frame[32][65] = '@';

frame[33][65] = '@';

frame[34][65] = '@';

frame[35][65] = '@';

frame[36][65] = '@';

frame[37][65] = '@';

frame[11][62] = '@';

frame[11][63] = '@';

frame[11][64] = '@';

frame[11][65] = '@';

frame[12][61] = '@';

frame[13][61] = '@';

frame[14][61] = '@';

frame[15][61] = '@';

frame[16][61] = '@';

frame[17][62] = '@';

frame[17][63] = '@';

frame[17][64] = '@';

frame[17][65] = '@';

for (int i = 0; i < 10; i++)

for (int j = 0; j < 20; j++)

if (i == 0 || j == 0 || i == 9 || j == 19)

frame[10 + ((selection) / 2) \* 20 + i][20 + ((selection) % 2) \* 40 + j] = char(219);

}

}

void printer()

{

system("cls");

cout<<"\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\t\t\t\tPRODUCTIVITY INTERFACE\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

for (int i = 10; i < 40; i++)

{

for (int j = 0; j < 100; j++)

cout << frame[i][j];

cout << endl;

}

}

int main()

{

int input = 0;

char a;

while (true)

{

first\_screen(input);

system("cls");

printer();

a = getch();

if (a == 'w'){

if(input>1)

input -= 2;

}

else if (a == 's'){

if(input<2)

input += 2;

}

else if (a == 'a'){

if(input%2>0)

input -= 1;

}

else if (a == 'd'){

if(input%2<1)

input += 1;

}

else if (a == ' ')

{

system("CLS");

return input;

}

system("CLS");

}

}

};

int main()

{

selection\_screen ob1;

while (true)

{

switch (ob1.main())

{

case 0:

mainr();

break;

case 1:

main3();

break;

case 2:

maint();

break;

case 3:

main1();

break;

}

}

}

**reflex.cpp**

#include <iostream>

//#include <stdlib.h>

#include <windows.h>

#include <chrono>

//#include <thread>

#include <conio.h>

//#include <time.h>

#include <unistd.h>

using namespace std;

using std::chrono::duration\_cast;

using std::chrono::milliseconds;

using std::chrono::seconds;

using std::chrono::system\_clock;

//char frame[50][100];

char random(){

return (char)(97+rand()/1310);

}

void print(char a){

if(a=='a'){

cout<<" @ \n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<" @@@@@@@ \n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<"@ @\n";

}

else if(a=='b'){

cout<<"@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@@@@@ \n";

}

else if(a=='c'){

cout<<" @@@@@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<" @@@@@ \n";

}

else if(a=='d'){

cout<<"@@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@@@@@@ \n";

}

else if(a=='e'){

cout<<"@@@@@@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@@@@@@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@@@@@@ \n";

}

else if(a=='f'){

cout<<"@@@@@@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@@@@@@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

}

else if(a=='g'){

cout<<" @@@@@ \n";

cout<<"@ @ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ @@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<" @@@@@ \n";

}

else if(a=='h'){

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

}

else if(a=='i'){

cout<<"@@@@@@@ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<"@@@@@@@ \n";

}

else if(a=='j'){

cout<<"@@@@@@@ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ @ \n";

cout<<" @@ \n";

cout<<" \n";

}

else if(a=='k'){

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

}

else if(a=='l'){

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@@@@@@@ \n";

}

else if(a=='m'){

cout<<"@@ @@ \n";

cout<<"@ @ @ @ \n";

cout<<"@ @ @ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

}

else if(a=='n'){

cout<<"@@ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @@ \n";

}

else if(a=='o'){

cout<<" @@@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<" @@@@@@@ \n";

}

else if(a=='p'){

cout<<"@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@@@@@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<"@ \n";

}

else if(a=='q'){

cout<<" @@@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ @ \n";

cout<<"@ @ @ \n";

cout<<" @@@@@@@ \n";

}

else if(a=='r'){

cout<<"@@@@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@@@@@ \n";

cout<<"@@ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

}

else if(a=='s'){

cout<<" @@@@@@@ \n";

cout<<"@ \n";

cout<<"@ \n";

cout<<" @@@@@@@ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @@@@@@@ \n";

}

else if(a=='t'){

cout<<"@@@@@@@ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

}

else if(a=='u'){

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<" @@@@@@@ \n";

}

else if(a=='v'){

cout<<"@ @\n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<" @ \n";

cout<<" \n";

}

else if(a=='w'){

cout<<"@ @ \n";

cout<<"@ @ \n";

cout<<" @ @ \n";

cout<<" @ @ @ \n";

cout<<" @ @ @ @ \n";

cout<<" @ @ @ @ \n";

cout<<" @ @ \n";

cout<<" \n";

}

else if(a=='x'){

cout<<"@ @\n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<" @@ \n";

cout<<" @@ \n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<"@ @ \n";

}

else if(a=='y'){

cout<<" @ @\n";

cout<<" @ @ \n";

cout<<" @ @ \n";

cout<<" @@ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

}

else if(a=='z'){

cout<<" @@@@@@@@ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @ \n";

cout<<" @@@@@@@@@ \n";

}

}

int mainr(){

srand(time(0));

char ch,ch2;

int time=0,score=0;

cout<<"Press a key when ready\n";

getch();

sleep(1);

cout<<"start\n";

int flag=1;

while(flag){

system("CLS");

ch=random();

//cout<<ch<<"\n";

print(ch);

time = duration\_cast<milliseconds>(system\_clock::now().time\_since\_epoch()).count();

ch2=getch();

time-=duration\_cast<milliseconds>(system\_clock::now().time\_since\_epoch()).count();

time=-time;

if(time>2000||ch!=ch2){

flag=0;

break;

}

else{

score++;

continue;

}

}

cout<<"\nScore is "<<score;

getch();

return 0;

}

**sclc.cpp**

#include <iostream>

#include <cmath>

#include <conio.h>

using namespace std;

void sum()

{

int x, y, sum;

cout << "\nenter first number\n";

cin >> x;

cout << "\nenter second number\n";

cin >> y;

sum = x + y;

cout << "sum is" << sum << endl;

}

void product()

{

int x, y, product;

cout << "\nenter first number\n";

cin >> x;

cout << "\nenter second number\n";

cin >> y;

product = x \* y;

cout << "product is" << product << endl;

}

void sub()

{

int x, y, sub;

cout << "\nenter first number\n";

cin >> x;

cout << "\nenter second number\n";

cin >> y;

sub = x - y;

cout << "subtraction is" << sub << endl;

}

void div()

{

int x, y, div;

cout << "\nenter first number\n";

cin >> x;

cout << "\nenter second number\n";

cin >> y;

div = x / y;

cout << "division is" << div << endl;

}

void remainder()

{

int x, y, remainder;

cout << "\nenter first number\n";

cin >> x;

cout << "\nenter second number\n";

cin >> y;

remainder = x \* y;

cout << "remainder is" << remainder << endl;

}

void square()

{

int x, square;

cout << "\nenter number\n";

cin >> x;

square = x \* x;

cout << "square is" << square << endl;

}

void cube()

{

int x, cube;

cout << "\nenter number\n";

cin >> x;

cube = x \* x \* x;

cout << "cube is" << cube << endl;

}

void cos()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = cos(x \* PI / 180);

cout << "cosine of " << x << " is " << result << endl;

}

void sin()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = sin(x \* PI / 180);

cout << "sine of " << x << " is " << result << endl;

}

void tan()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = tan(x \* PI / 180);

cout << "tan of " << x << " is " << result << endl;

}

void cosh()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = cosh(x);

cout << "hyperbolic cosine of " << x << " is " << result << endl;

}

void sinh()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = sinh(x);

cout << "hyperbolic sine of " << x << " is " << result << endl;

}

void tanh()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = tanh(x);

cout << "hyperbolic tangent of " << x << " is " << result << endl;

}

void acos()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = acos(x) \* 180.0 / PI;

cout << "arc cosine of " << x << " is " << result << endl;

}

void asin()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = asin(x) \* 180.0 / PI;

cout << " arc sine of " << x << " is " << result << endl;

}

void atan()

{

double x, result, PI = 3.14159265;

cout << "\nenter angle\n";

cin >> x;

result = tanh(x);

cout << "arc tangential of " << x << " is " << result << endl;

}

void exp()

{

double x, result;

cout << "\nenter number\n";

cin >> x;

result = exp(x);

cout << "exponential value of " << x << " is = " << result << endl;

}

void log()

{

double x, result;

cout << "\nenter value to take x log \n";

cin >> x;

result = log(x);

cout << " logarithm of " << x << " is " << result << endl;

}

void log10()

{

double x, result;

cout << "\nenter value to take x log \n";

cin >> x;

result = log10(x);

cout << "common logarithm of " << x << " is " << result << endl;

}

void atan2()

{

float x, y, result, PI = 3.14;

cout << "\nenter first number\n";

cin >> x;

cout << "\nenter second number\n";

cin >> y;

result = atan2(x, y) \* 180 / PI;

cout << "arc tangent for " << x << " and " << y << " is " << result << endl;

}

void pow()

{

int x, y, pow1;

cout << "\nenter number\n";

cin >> x;

cout << "\nenter power\n";

cin >> y;

pow1 = pow(x, y);

cout << "\n"

<< x << "raise to power" << y << "is" << pow1;

}

int main3()

{ a:

system("cls");

cout << "SCIENTIFIC CALCULATOR" << endl;

long func;

cout << "Enter the function you want to perform(+,\*,D,-,R,S,C,c,s,t,h,H,T,A,y,d,E,L,l,N,P): \n";

cout << "Key :\n";

cout << "`+' is for addition\n";

cout << "`\*' is for multiplication\n";

cout << "`D' is for division\n";

cout << "`-' is for subtraction \n";

cout << "`R' is for remainder\n";

cout << "`S' is for square\n";

cout << "`C' is for cube\n";

cout << "`c' is for cosine\n";

cout << "`s' is for sine\n";

cout << "`t' is for tangent\n";

cout << "`T' is for hyperbolic tangent\n";

cout << "`h' is for hyperbolic cosine\n";

cout << "`H' is for hyperbolic sine\n";

cout << "`A' is for arc cosine\n";

cout << "`y' is for arc sine\n";

cout << "`d' is for arc tangential\n";

cout << "`E' is for exponent\n";

cout << "`L' is for logarithm\n";

cout << "`l' is for common logarithm i.e log10\n";

cout << "`N' is for atan2\n";

cout << "`P' is for power\n";

func = getche();

if (func == '+')

{

sum();

cout << "\*\*\*\n";

}

else if (func == '\*')

{

product();

cout << "\*\*\n";

}

else if (func == '-')

{

sub();

cout << "\*\*\n";

}

if (func == 'D')

{

div();

cout << "\*\n";

}

else if (func == 'R')

{

remainder();

cout << "\*\*\*\n";

}

else if (func == 'S')

{

square();

cout << "\*\n";

}

else if (func == 'C')

{

cube();

cout << "\*\*\n";

}

else if (func == 'c')

{

cos();

cout << "\*\*\n";

}

else if (func == 's')

{

sin();

cout << "\*\*\*\n";

}

else if (func == 't')

{

tan();

cout << "\*\*\n";

}

else if (func == 'h')

{

cosh();

cout << "\*\*\n";

}

else if (func == 'H')

{

sinh();

cout << "\*\*\n";

}

else if (func == 'T')

{

tanh();

cout << "\*\*\n";

}

else if (func == 'A')

{

acos();

cout << "\*\n";

}

else if (func == 'y')

{

asin();

cout << "\*\*\*\n";

}

else if (func == 'd')

{

atan();

cout << "\*\*\*\n";

}

else if (func == 'E')

{

exp();

cout << "\*\*\n";

}

else if (func == 'L')

{

log();

cout << "\*\*\n";

}

else if (func == 'l')

{

log10();

cout << "\*\n";

}

else if (func == 'N')

{

atan2();

cout << "\*\n";

}

else if (func == 'P')

{

pow();

cout << "\*\*\n";

}

int i=0;

cout<<"to do more operation press 1\nto end calculator operation press 2\n";

cin>>i;

if(i==1)

goto a;

else

return 0;

}

**td.cpp**

#include <iostream>

#include <vector>

#include <fstream>

#include <conio.h>

#include <windows.h>

using namespace std;

class t

{

public:

string task, deadline;

void input()

{

cout << "Enter task name ";

cin >> task;

cout << "Enter task deadline ";

cin >> deadline;

}

};

void add()

{

t temp;

temp.input();

fstream f1;

f1.open("file.txt", ios::app);

f1 << "\n"

<< temp.task << " " << temp.deadline << " N";

f1.close();

}

void show()

{

cout << "Sno.\tTask name \tTask deadline\tstatus\n";

string s;

fstream f2;

f2.open("file.txt");

int i = 1;

while (!f2.eof())

{

f2 >> s;

if (s.length() < 3)

break;

cout << i++ << "\t ";

cout << s << "\t\t";

f2 >> s;

cout << s << "\t\t";

f2 >> s;

if (s.compare("N") == 0)

cout << "Not done";

else if (s.compare("D") == 0)

cout << "Done";

cout << "\n";

}

}

void del(int i)

{

fstream f1;

f1.open("file.txt");

vector<string> ob;

// int j=0;

f1.seekg(0, ios::beg);

while (f1)

{

string s, temp;

f1 >> temp;

s = s + temp + " ";

f1 >> temp;

s = s + temp + " ";

f1 >> temp;

s = s + temp + "\n";

if (s.length() > 3)

ob.push\_back(s);

}

ob.erase(ob.begin() + i);

f1.close();

f1.open("file.txt", ios::out | ios::trunc);

for (int k = 0; k < ob.size(); k++)

{

f1 << ob[k];

}

f1.close();

}

void edit()

{

fstream f1;

f1.open("file.txt");

vector<string> ob;

f1.seekg(0, ios::beg);

while (f1)

{

string s, temp;

f1 >> temp;

s = s + temp + " ";

f1 >> temp;

s = s + temp + " ";

f1 >> temp;

s = s + temp + "\n";

if (s.length() > 3)

ob.push\_back(s);

}

f1.close();

cout << "Enter task no. to edit ";

int op;

cin >> op;

system("cls");

cout << "Enter 1 to edit task \n";

cout << "Enter 2 to edit task status\n";

int i;

cin >> i;

f1.open("file.txt");

string s, temp;

if (i == 1)

{

t temp;

temp.input();

s = temp.task + " " + temp.deadline + " " + "N" + "\n";

ob[op - 1] = s;

f1.close();

f1.open("file.txt", ios::out | ios::trunc);

// show();

for (int k = 0; k < ob.size(); k++)

{

f1 << ob[k];

}

f1.close();

}

else if (i == 2)

{

f1.close();

cout << "Enter 1 to mark as done ";

cin >> s;

//s="";

if (s.compare("1") == 0)

{

s="";

int ab=0,c=0;;

while(c<2){

if((ob[op - 1])[ab]==' ')

c++;

s=s+(ob[op - 1])[ab++];

}

s=s+"D\n";

//Sleep(2);

ob[op-1]=s;

f1.open("file.txt", ios::out | ios::trunc);

for (int k = 0; k < ob.size(); k++)

{

f1 << ob[k];

}

f1.close();

}

}

}

int maint()

{

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << " || ||" << endl;

cout << " || WELCOME TO YOUR DAILY TASK MANAGER ||" << endl;

cout << " || WRITE THEM DOWN ||" << endl;

cout << " || IMPLIMENT THEM ||" << endl;

cout << " || KEEP CALM ||" << endl;

cout << " ||&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&||" << endl;

cout << " || ||" << endl;

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "======================= TO DO LIST =============================\n";

cout << "================================================================\n";

getch();

a:

cout << "Enter 1 to add\n";

cout << "Enter 2 to show\n";

cout << "Enter 3 to delete\n";

cout << "Enter 4 to edit\n";

cout << "Enter 5 to exit\n";

int input;

cin >> input;

system("cls");

if (input == 1)

{

add();

}

else if (input == 2)

{

show();

}

else if (input == 3)

{

int a;

cout << "Enter task number to delete ";

cin >> a;

del(a - 1);

}

else if (input == 4)

{

edit();

}

else if (input == 5)

goto b;

goto a;

b:

return 0;

}

**timerup.cpp**

#include <iostream>

//#include <stdlib.h>

#include <windows.h>

#include <chrono>

#include <thread>

#include <conio.h>

//#include <time.h>

//#include <unistd.h>

#include <thread>

#include <cstdio>

using namespace std;

/\*using std::chrono::duration\_cast;

using std::chrono::milliseconds;

using std::chrono::seconds;

using std::chrono::system\_clock;\*/

int input = 100;

char i = '0';

void print\_time(int time)

{

cout << "\nIf you want to stop the timer press '1'\nif you want to add extra time by 1 minute press '2'\nif you want to reduce the time by 1 minute press '3'\n";

cout << "\t\t\t\t\t\t\t" << time / 3600 << ":" << (time % 3600) / 60 << ":" << time % 60;

}

void timer()

{

while (input + 1 > 0)

{

if (input == -1)

return;

system("CLS");

print\_time(input);

cout << endl;

Sleep(600);

input--;

if (input < 6)

{

Beep(8000, 400);

}

else

Sleep(400);

}

if (input == -1)

cout << "timer has run out press 4 to continue or 5 to make a new timer";

return;

}

int main1()

{

b:

int time = 0;

cout << "you will have 4 inbuilt timer options:" << endl;

cout << "If you want to stop the timer press '1'\nif you want to add extra time by 1 minute press '2'\nif you want to reduce the time by 1 minute press '3'\n";

cout << "1 for brush teeth:00:02:00\n2 for face mask: 00:15:00\n3 for boil eggs: 00:10:00\n4 for custom timer \n";

int n;

cin >> n;

thread t;

string x;

// try

//{

switch (n)

{

case 1:

cout << "brush teeth\n";

input = 120;

// cout<<"\t"<<input;

t = thread(timer);

break;

case 2:

cout << "face mask\n";

input = 900;

t = thread(timer);

break;

case 3:

cout << "boil eggs\n";

input = 600;

t = thread(timer);

break;

case 4:

// string x;

cout << "enter the name of custom timer: ";

fflush(stdin);

cin >> x;

fflush(stdin);

int h, m, s;

cout << "Enter timer in hours minutes and seconds :\n";

cin >> h >> m >> s;

try

{

if (m > 60)

{

throw 1;

}

}

catch (int x)

{

cout << "minutes cannot be greater than 60 minutes, exception run" << endl;

// exit(0);

}

try

{

if (s > 60)

{

throw 1;

}

}

catch (int x)

{

cout << "seconds canot be greater than 60 seconds, exception run" << endl;

// exit(0);

}

input = h \* 3600 + m \* 60 + s;

t = thread(timer);

cout << "herer";

break;

default:

cout << "please enter right operation";

}

a:

char i = getch();

if (i == '1')

{

cout << "ok the timer has been stopped";

input = -2;

t.join();

return 0;

}

else if (i == '2')

{

input += 60;

goto a;

}

else if (i == '3')

{

if (input <= 60)

{

system("CLS");

cout << "0:0:0";

cout << "\ntimer has been run out";

input = 0;

t.detach();

return 0;

}

input -= 60;

goto a;

}

else if (i == '4' && input != -1)

{

cout << "timer has run out";

}

else if (i == '5')

{

t.detach();

system("cls");

fflush(stdin);

goto b;

}

// cout << "timer has run out";

t.join();

return 0;

}

**Code Snippets**

















