**Міністерство освіти та науки України**
**Національний технічний університет України**
**“Київський політехнічний інститут імені Ігоря Сікорського”**
**Факультет прикладної математики**
**Кафедра системного програмування і спеціалізованих комп’ютерних систем**

# Домашня контрольна робота
## з дисципліни
## “ Програмування”

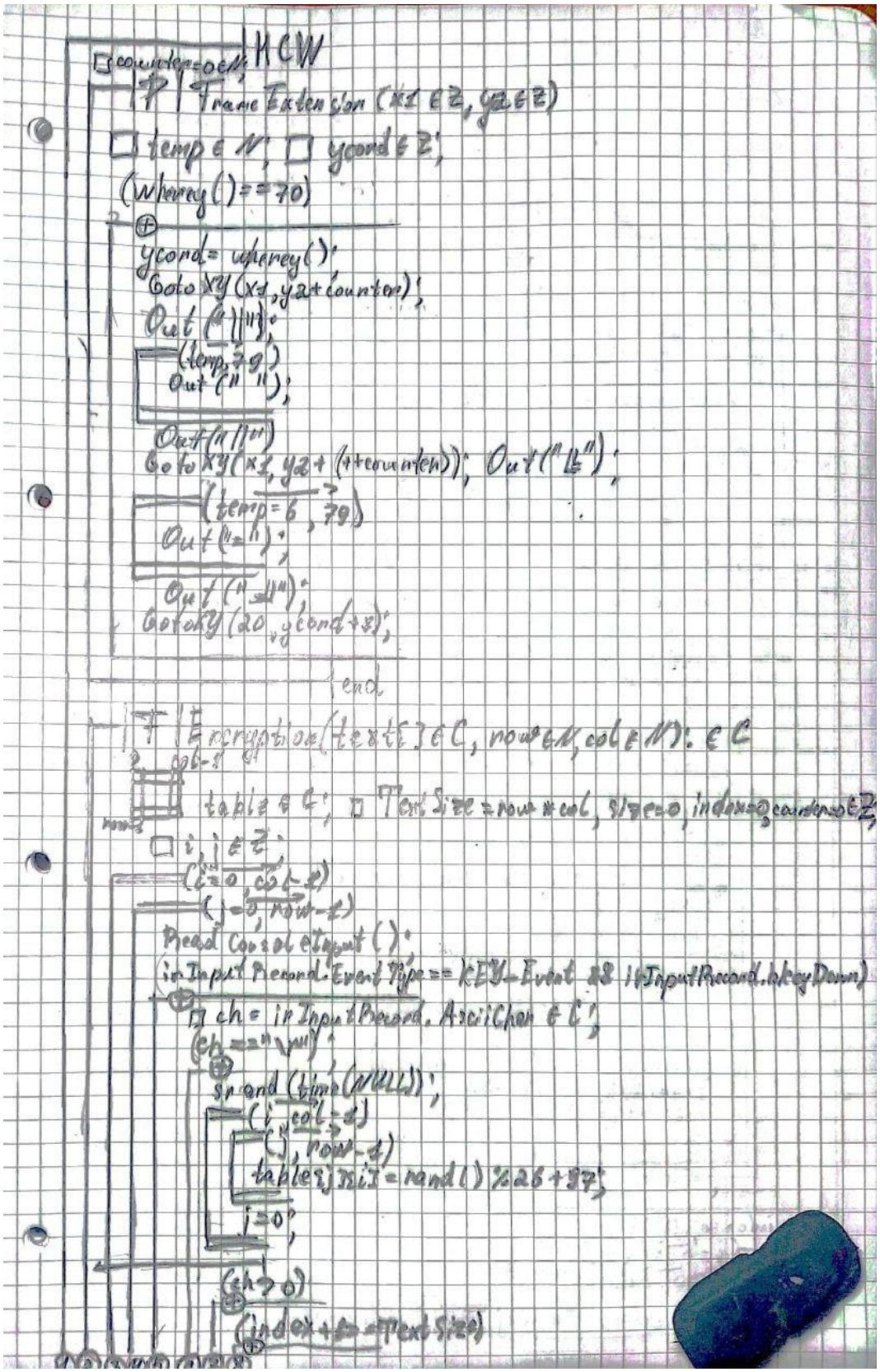
**Виконав: Фесенко Д.О.**
**Студент групи КВ-34**
**Варіант №23**

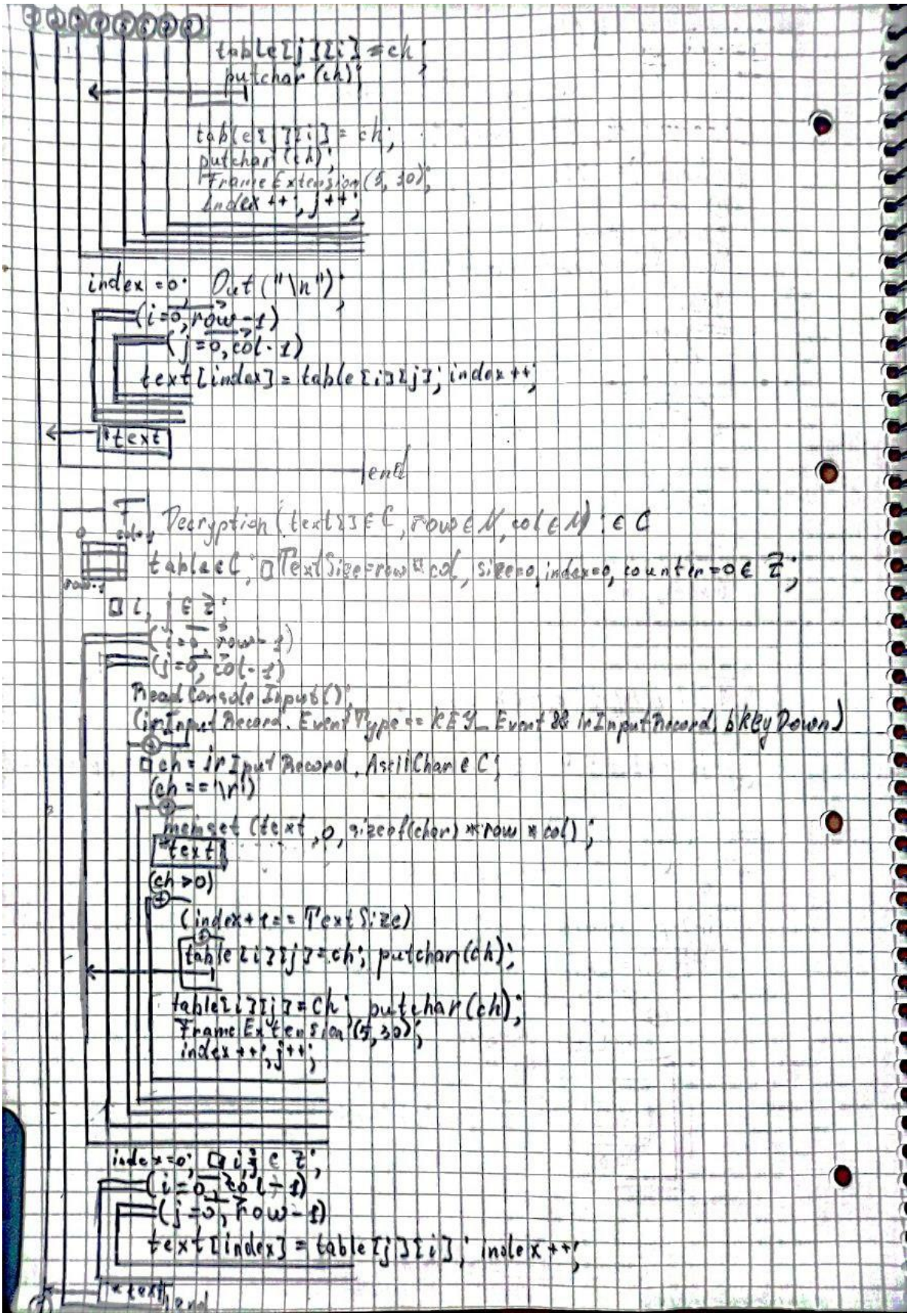
**Київ 2023**

## Постановка задачі

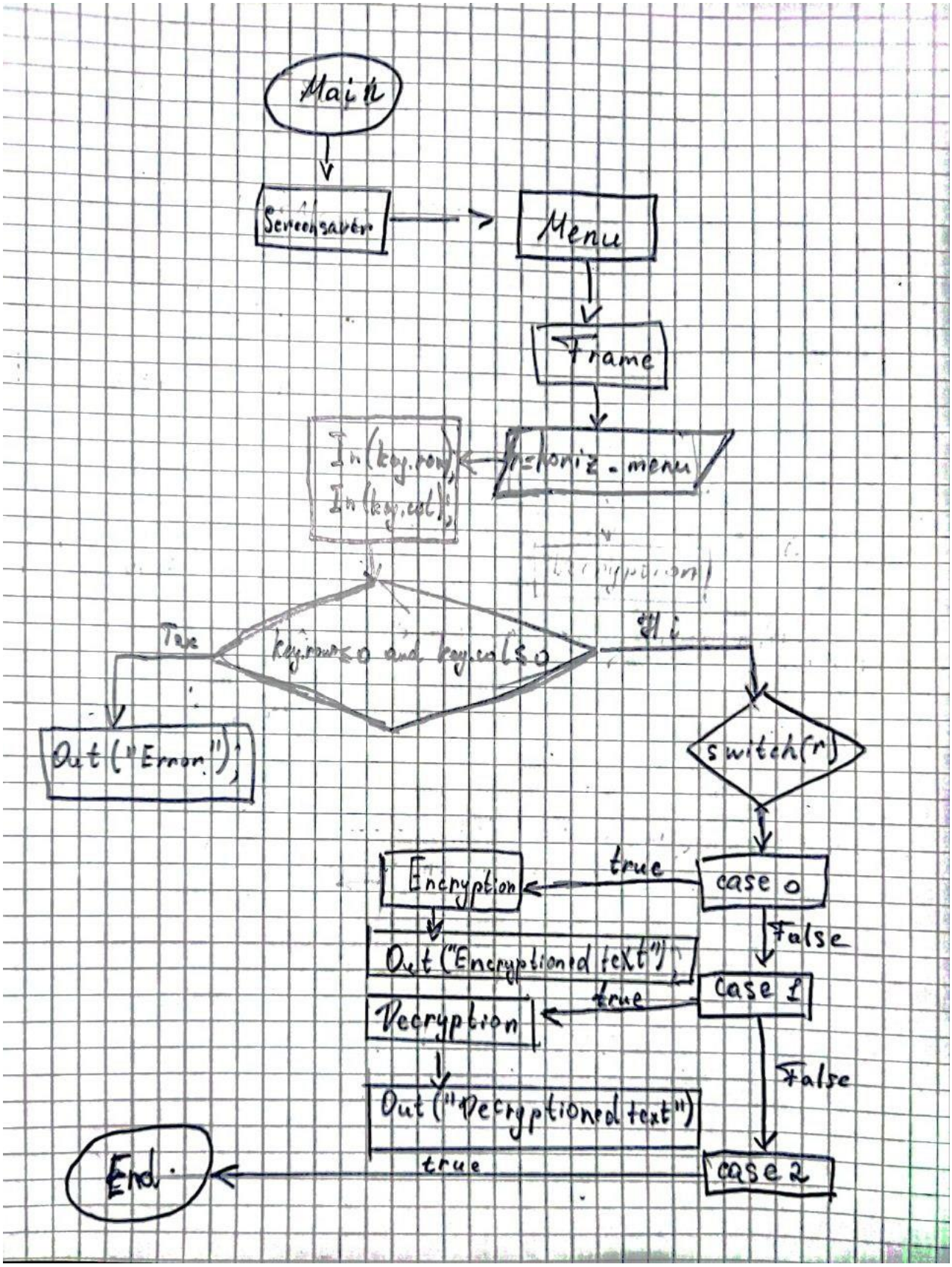
Заданий текст. Розробити табличний спосіб шифровки текста, зашифрувати його, розшифрувати текст.

# Діаграми дій




table[j][i] = ch;
putchar(ch);
table[i][j] = ch;
putchar(ch);
FrameExtension(5, 30);
index++; j++;
index = 0; Out("\n");
(i=0, row-1)
(j=0, col-1)
text[index] = table[i][j]; index++;
*text
end
Decryption(text[] ∈ C, row ∈ N, col ∈ N) : ∈ C
table ∈ C; □ TextSize=row*col, size=0, index=0, counter=0 ∈ Z;
□ i, j ∈ Z;
ReadConsoleInput();
(inInputRecord.EventType == KEY_EVENT && inInputRecord.bKeyDown)
□ ch = inInputRecord.AsciiChar ∈ C;
(ch == '\r')
memset(text, 0, sizeof(char) * row * col);
*text
(ch > 0)
(index++ == TextSize)
table[i][j] = ch; putchar(ch);
table[i][j] = ch; putchar(ch);
FrameExtension(5, 30);
index++; j++;
index = 0; □ i, j ∈ Z;
(i=0, col-1)
(j=0, row-1)
text[index] = table[j][i]; index++;
*text
end

## Ілюстрація взаємозв'язків блоків в програмі



# Код програми

```
#include <windows.h>
#include <stdio.h>
#include<time.h>
#include<string.h>
HANDLE hStdin;
HANDLE hStdOut;
COORD position={ 0, 0 };
DWORD fdwSaveOldMode;
INPUT_RECORD irInputRecord;
DWORD cNumRead;
#define CoItems 3
int  counter = 0;//Глобальна через потребу зміни значень лічильника в залежності від функції
void SetColor(int color);//
void GotoXY(int X, int Y);//
void Frame(int x1, int y1, int x2, int y2);//
void Screensaver();//
int horiz_menu(int k2, int kp);//
void Menu_main();
char Encryption(char text[], int row, int col);
char Decryption(char text[], int row, int col);
void FrameExtension(float x1, float y2);
//Робота програми також основана на приємному з боку
//користувача вигляді(під час вводу текст може вийти за рамки)
int main() {
hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
hStdin = GetStdHandle(STD_INPUT_HANDLE);
   Screensaver();
   Menu_main();
   SetConsoleMode(hStdin, ENABLE_PROCESSED_INPUT);

   return 0;
}

void GotoXY(int X, int Y) {
   position.X = X;
   position.Y = Y;
   SetConsoleCursorPosition(hStdOut, position);
   /*SetConsoleCursorPosition(hConsole, coord);*/  //функція переміщення курсора по X і У
}

int wherex() {
   CONSOLE_SCREEN_BUFFER_INFO info;
   GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_HANDLE), &info);
   return info.dwCursorPosition.X;
}

int wherey() {
   CONSOLE_SCREEN_BUFFER_INFO info;
   GetConsoleScreenBufferInfo(GetStdHandle(STD_OUTPUT_HANDLE), &info);
   return info.dwCursorPosition.Y;
}

void SetColor(int color) {
   HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
   SetConsoleTextAttribute(hConsole, color);
}

void Screensaver() {
   SetColor(9);//<-----Color changing
   system("cls"); Frame(5, 5, 80, 22);
```

```
    GotoXY(25, 8);  printf("Work \"Enigma\" created by Denis Fesenko");
    GotoXY(32, 10); printf("Student's of group: KV-34");
    GotoXY(39, 12); printf("Variant 23");
    GotoXY(40, 20); printf("Kyiv 2023");
    GotoXY(65, 21); printf("[Press any key]");
    _getch(); system("cls");//<-clear screen
    return;
} // Screensaver()

void Frame(int x1, int y1, int x2, int y2) {
    SetColor(9);
            int i;
            GotoXY(x1, y1); printf("\311");//'╔'
            for (i = (x1 + 1); i <= (x2 - 1); i++) printf("\315");// '═'

    printf("\273");//'╗'
            for (i = (y1 + 1); i <= (y2 - 1); i++) {
                    GotoXY(x1, i); printf("\272");//'║'
                    GotoXY(x2, i); printf("\272");//'║'
            }
            GotoXY(x1, y2); printf("\310");// '╚'

            for (i = (x1 + 1); i <= (x2 - 1); i++)
                    printf("\315");// '═'
            printf("\274");// '╝'
            return;
} //Frame()

void Menu_main() {
    short regime;
    struct { int row, col; }key;
    while (1) {
        system("cls");
        Frame(5, 5, 80, 20);
        //Frame(37, 6, 48, 8);
        GotoXY(40, 7); printf("ENIGMA");
        GotoXY(6, 9);
        for (int i = 1; i < 75; i++) {
            printf("*");
        }
        regime = horiz_menu(18, CoItems);
        counter = 0;//Обнуление счетчика расширяемых строк, ибо без обнуления стрОки расширялись начиная
каждый раз со старой строки
        switch (regime) {
        case 0: {
            system("cls");
            Frame(5, 5, 80, 30);
            Frame(27, 7, 62, 9);
            GotoXY(32, 8);

            printf("Input key please(table size):");
            GotoXY(6, 6); printf("[And press Enter]");
            GotoXY(20, 11);
            scanf("%d", &key.row);
            GotoXY(20, 12);
            scanf("%d", &key.col);

            if (key.row <= 0 || key.col <= 0) {//Розмір текста 0
                Frame(27, 14, 62, 17);
                GotoXY(29, 15);
                printf("The text cannot be empty,");
                GotoXY(38, 16); printf("otherwise why all this?");
```

```
        GotoXY(6, 6); printf("[Press Enter to exit]            ");
        char ch;
        do {
            ch = _getch();
        } while (ch != '\r');
        break;
    }

    char* text = (char*)malloc(sizeof(char) * key.row*key.col);
    Frame(27, 14, 62, 16);

    GotoXY(32, 15);
    printf("Input text: ");
    GotoXY(6, 6); printf("[And press Enter to save the text]");
    position.X = 20; position.Y = 18;
    SetConsoleCursorPosition(hStdOut, position);

    Encryption(&text[0],key.row,key.col);

    position.Y= wherey();
    Frame(27, position.Y+1, 62, position.Y+3);
    GotoXY(32, position.Y -1);
    printf("Encrypted text:");
    GotoXY(20, position.Y+3);
    for (int i = 0; i < key.row * key.col; i++) {
        printf("%c", text[i]);
        FrameExtension(5,30);
    }
    GotoXY(6, 6); printf("[Press Enter to exit]            ");
    char ch;
    do {  ch =_getch();
    } while (ch!='\r');
    free(text);
    break; }
case 1: {

    system("cls");
    Frame(5, 5, 80, 30);
    Frame(27, 7, 62, 9);
    GotoXY(30, 8);

    printf("Input key please(table size) :");
    GotoXY(6, 6); printf("[And press Enter]");
    GotoXY(20, 11);
    scanf("%d", &key.row);
    GotoXY(20, 12);
    scanf("%d", &key.col);

    if (key.row <=0 || key.col <= 0) {//Розмір текста 0
        Frame(27, 14, 62, 17);
        GotoXY(29, 15);
        printf("The text cannot be empty,");
        GotoXY(38, 16); printf("otherwise why all this?");
        GotoXY(6, 6); printf("[Press Enter to exit]            ");
        char ch;
        do {
            ch = _getch();
        } while (ch != '\r');
        break;
    }
    char* text = (char*)malloc(sizeof(char) * key.row * key.col);
    Frame(27, 14, 62, 16);
```

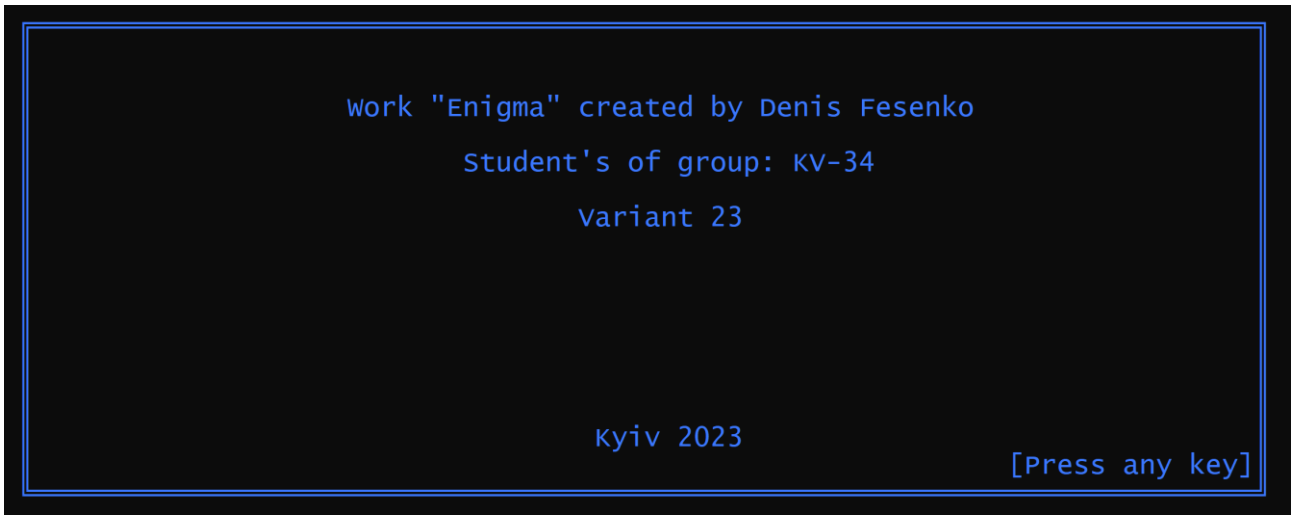
```
        GotoXY(39, 15);
        printf("Input text : ");
        GotoXY(6, 6); printf("[And press Enter to save the text]");

        GotoXY(20, 18);
        Decryption(&text[0], key.row, key.col);

        position.Y = wherey();
        Frame(27, position.Y + 2, 62, position.Y + 4);
        GotoXY(34, position.Y-1);
        if (strlen(text) == 0) {//Ввели невірний ключ, і для тих, хто не знає ключ - немає доступ до тексту!
        printf("Entered the wrong key(");
        }
        else {
        printf("   Decrypted text :");
          GotoXY(20, position.Y + 3);

          for (int i = 0; i < key.row * key.col; i++) {
            printf("%c", text[i]);
            FrameExtension(5, 30);
          }
        }
        GotoXY(6, 6); printf("[Press Enter to exit]            ");
        char ch;
        do {
          ch = _getch();
        } while (ch != '\r');

        free(text);
        break;
      }
      case 2: {
        system("cls");
        return 0; }
      } //switch
    } //while
}

int horiz_menu(int k2, int kp) {
    typedef char punkt[30];
    punkt m[CoItems] = { "Encryption", "Decryption", "Exit" };

    int i, t, r, result;
    char ch; int Flag = 1;
    int pp[CoItems];
    t = strlen(m[0]);  r = 0;
    for (i = 0; i < kp; i++)  r += strlen(m[i]);
    r = ((80 - r) / kp) - 1;
    Frame(6, k2 - 1, 79, k2 + 1);
    GotoXY(25, 15);
    printf("Choose with <- and -> and press ENTER");
    GotoXY((r / 2 + 1), k2);
    for (i = 0; i < kp; i++) {
      if (i == 0) { SetColor(7); }
      else { SetColor(9); }
      GotoXY(wherex() + 11, wherey());
      pp[i] = wherex();
      printf("%s", m[i]);
      if (i == kp) r = 3;
    }
    i = 0;
```

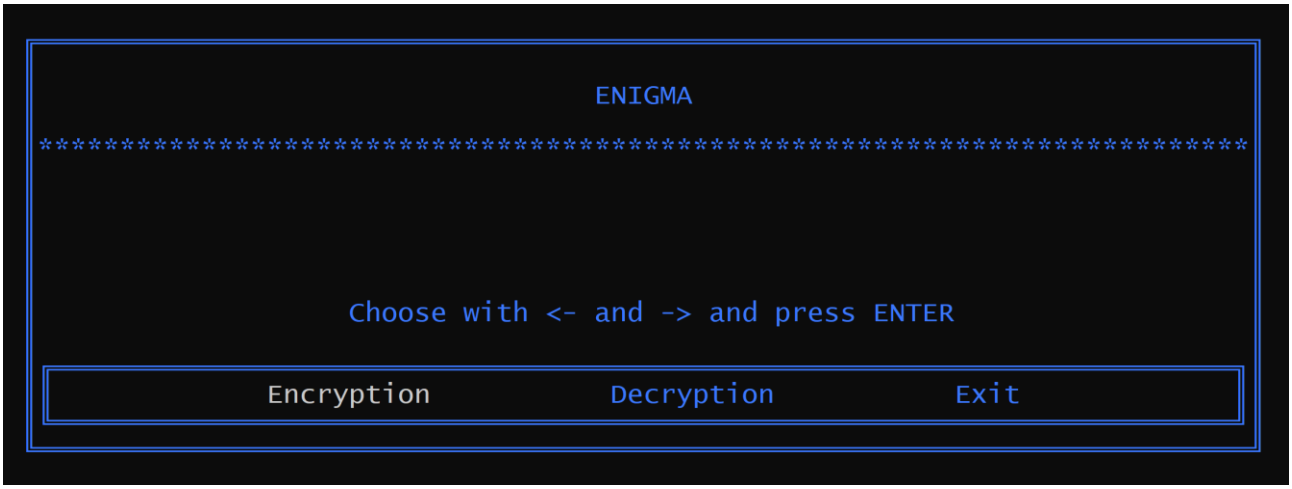
```
   while (Flag == 1) {//Moving between items
      ch = _getch();
      switch (ch) {
      case 77: {//-->
         GotoXY(pp[i], k2);
         SetColor(9);
         printf("%s", m[i]);
         i++;
         if (i == kp) {
            i = 0;
            SetColor(9);
            GotoXY(pp[0], k2);
            SetColor(7);
            printf("%s", m[i]);
         }
         GotoXY(pp[i], k2);
         SetColor(7);
         printf("%s", m[i]);
          break;
      };//case 'q'
      case 75: {//<--

         GotoXY(pp[i], k2);
         SetColor(9);
         printf("%s", m[i]);
         if (i == 0) {
         i = kp-1;
         GotoXY(pp[i], k2);
         SetColor(7);
         printf("%s", m[i]);
          break;
         }
         i--;
         GotoXY(pp[i], k2);
         SetColor(7);
         printf("%s", m[i]);
          break;
      }
      case 13: {//Enter
         Flag = 0;
         result = i;
      }//case '13-enter'

      };// switch
   }// while
   return result;
} //goriz_menu()

char Encryption(char text[], int row, int col) {
   char** table = (char**)malloc(sizeof(char*) * row);
   for (int i = 0; i < row; i++) {
      table[i] = (char*)malloc(sizeof(char) * col);
   }
   int TextSize = row * col;
   int size = 0; // Current number of characters
   int index = 0,counter=0;

   for (int i = 0; i < col;i++) { //Заповнюємо масив повністю, або до натискання Enter
      for (int  j= 0; j < row; ) {
         ReadConsoleInput(hStdin, &irInputRecord, 1, &cNumRead);//Зчитуємо будь-яку активність з консолі
         if (irInputRecord.EventType == KEY_EVENT && irInputRecord.Event.KeyEvent.bKeyDown) {//Відбулась
активність натискання клавіш
```

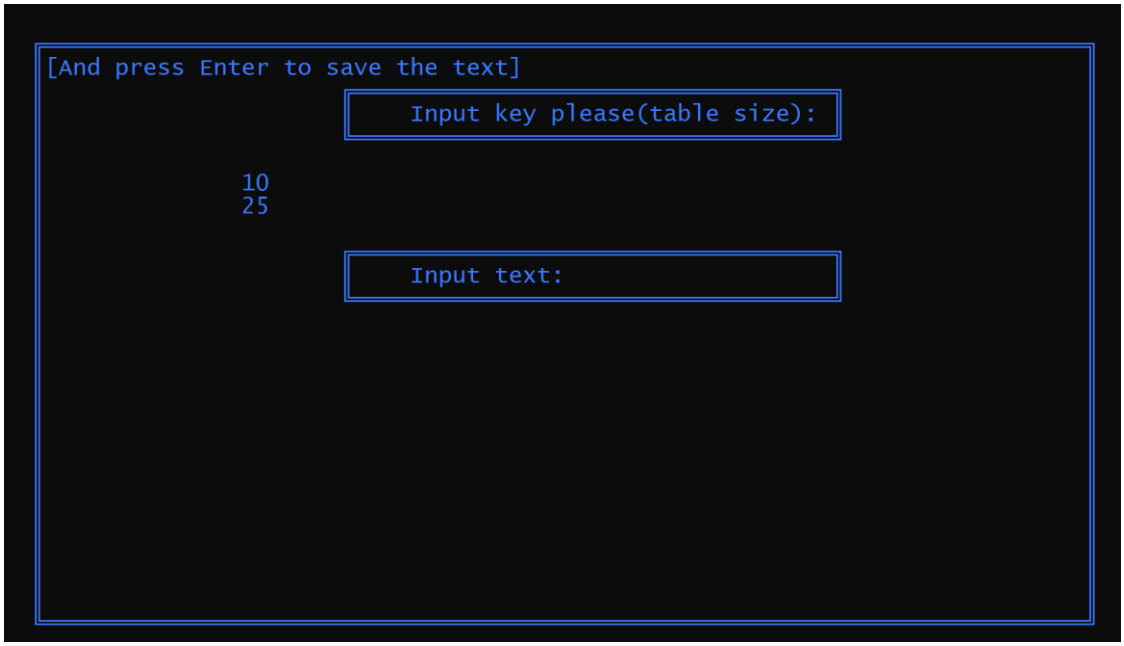
```
            char ch = irInputRecord.Event.KeyEvent.uChar.AsciiChar;// Отримуємо символ без виводу на консоль
            if (ch == '\r') { // При натисканні Enter(таке можливе лише при неповному заповнюванні масиву)
               srand(time(NULL));//Заповнюємо до кінця ранд
               for (; i < col; i++) {
                  for (; j < row; j++) {
                     table[j][i] = rand() % 26 + 97;
                  }
                  j = 0;
               }
               i = col; break; //Повний вихід с циклів
            }
            else if (ch>0) {//Перевірка активності на натискання б-я символа
               if (index+1 == TextSize) {//Останній символ
                  table[j][i] = ch;
                  putchar(ch);
                  break;
               }
               table[j][i] = ch;//Запис у масив та вивід у консоль
               putchar(ch);
               FrameExtension(5, 30);
               index++; j++;
            }
         }
      }
   }
   index = 0;
   printf("\n");
   for (int i = 0; i < row; i++) { //Запис у відповідний масив зашифрованого тексту
      for (int j = 0; j < col; j++) {
         text[index] = table[i][j];index++;
      }
   }

   for (int i = 0; i < row; i++) {
      free(table[i]);
   }
   free(table);
   return *text;
}
char Decryption(char text[], int row, int col) {
   char** table = (char**)malloc(sizeof(char*) * row);
   for (int i = 0; i < row; i++) {
      table[i] = (char*)malloc(sizeof(char) * col);
   }
   int TextSize = row * col;
   int size = 0; // Current number of characters
   int index = 0, counter = 0;

   for (int i = 0; i < row; i++) { //Заповнюємо масив повністю, або до натискання Enter
      for (int j = 0; j < col; ) {
         ReadConsoleInput(hStdin, &irInputRecord, 1, &cNumRead);//Зчитуємо будь-яку активність з консолі
         if (irInputRecord.EventType == KEY_EVENT && irInputRecord.Event.KeyEvent.bKeyDown) {//Відбулась
активність натискання клавіш
            char ch = irInputRecord.Event.KeyEvent.uChar.AsciiChar;// Отримуємо символ без виводу на консоль
            if (ch == '\r') { // При натисканні Enter(таке можливе лише при неповному заповнюванні масиву)
               memset(text, 0, sizeof(char) * row * col);//Обнулювання тексту
               for (int i = 0; i < row; i++) {
                  free(table[i]);
               }
               free(table);
               return *text;//Повертаємо обнульований текст для подальшого виведення повідомлення про
помилку
```

```
            }
            else if (ch > 0) {//Перевірка активності на натискання б-я символа
              if (index + 1 == TextSize) {//Останній символ
                table[i][j] = ch;
                putchar(ch);
                break;
              }
              table[i][j] = ch;//Запис у масив та вивід у консоль
              putchar(ch);
              FrameExtension(5, 30);
              index++; j++;
            }
          }
        }
      }
    index = 0;
    for (int i = 0; i < col; i++) { //Запис у відповідний масив розшифрованого тексту
      for (int j = 0; j < row; j++) {
        text[index] = table[j][i]; index++;
      }
    }
//kd riespnrur
    for (int i = 0; i < row; i++) {
      free(table[i]);
    }
    free(table);
    return *text;
}

void FrameExtension(float x1,float y2) {
    int temp = 6;
    float ycord;

    if (wherex() == 70) {//Розширенні рамки по Y при створюванні нового рядка

      ycord = wherey();
      GotoXY(x1, y2 + counter);
      printf("\272");//'║'
      for (; temp < 80; temp++) {
        printf(" ");
      }
      printf("\272");//'║'

      GotoXY(x1, y2 + ++counter);
      printf("\310"); // '╚'

      for (temp = 6; temp <= 79; temp++)
        printf("\315");// '═'
      printf("\274");// '╝'

      GotoXY(20, ycord + 1);

    }
}
```

# Результати

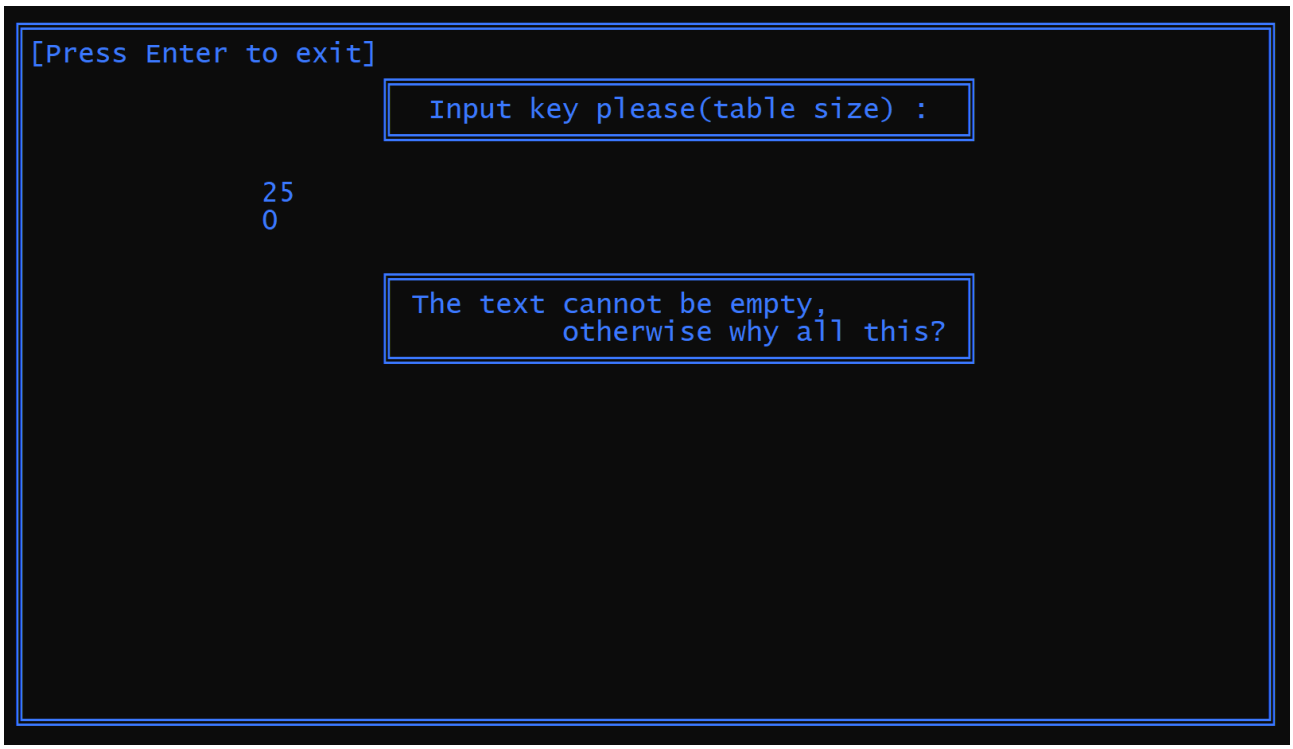

Work "Enigma" created by Denis Fesenko
Student's of group: KV-34
Variant 23
Kyiv 2023
[Press any key]

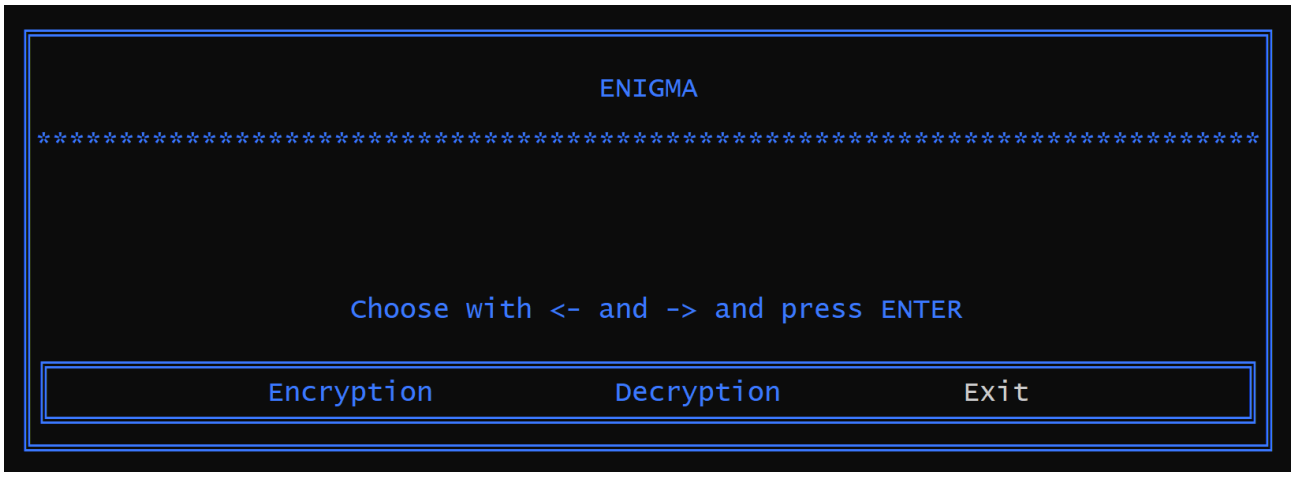

ENIGMA
Choose with <- and -> and press ENTER
Encryption
Decryption
Exit

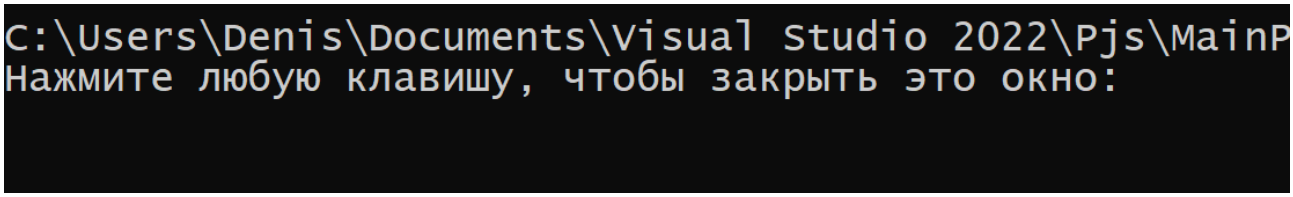

[And press Enter to save the text]
Input key please(table size):
10
25
Input text:


[Press Enter to exit]
Input key please(table size):
25
10
Input text:
The moon cast its silver glow upon the tranquil la
ke, as whispers of the night danced with the gentl
e breeze. Shadows intertwined with the shimmering
ripples, weaving a tale of serenity under the star
ry canvas. Amidst this serene symphony, dreams fou
Encrypted text:
Tgkgeirfrrhleh ni yeeo,tbeps n w  rdpecem ade lra
ousaeweensop nzisnvynowcet,iam nhe.h tspc id  wy.h
ats Ste  oshpwhhauAnteeiaevnmy  rtd idi,itshosned
tr  whgrsdsaotsi  tr nfh mat esq eim htaiut netehm
lihgtra isvleeeilss e  nrnet frlnttg aso ailw oreu


[Press Enter to exit]
Input key please(table size) :
25
10
Input text :
Tgkgeirfrrhleh ni yeeo,tbeps n w  rdpecem ade lra
ousaeweensop nzisnvynowcet,iam nhe.h tspc id  wy.h
ats Ste  oshpwhhauAnteeiaevnmy  rtd idi,itshosned
tr  whgrsdsaotsi  tr nfh mat esq eim htaiut netehm
lihgtra isvleeeilss e  nrnet frlnttg aso ailw oreu
Decrypted text :
The moon cast its silver glow upon the tranquil la
ke, as whispers of the night danced with the gentl
e breeze. Shadows intertwined with the shimmering
ripples, weaving a tale of serenity under the star
ry canvas. Amidst this serene symphony, dreams fou


[Press Enter to exit]
Input key please(table size) :
25
0
The text cannot be empty,
otherwise why all this?


ENIGMA
**********
Choose with <- and -> and press ENTER
Encryption
Decryption
Exit


C:\Users\Denis\Documents\Visual Studio 2022\Pjs\MainP
Нажмите любую клавишу, чтобы закрыть это окно: