```
#include <stdio.h>
#include <conio.h>

void box(int x,int y,int width,int height,int fc,int bc);
void DrawMenu(int,char **,int);

#define HB 205
#define VB 186
#define TRC 187
#define BRC 188
#define BLC 200
#define TLC 201
#define TBR 185
#define TBL 204

#define CLRSCR  window(1,1,80,25);textbackground(0);clrscr();
#define UP 72
#define DOWN 80

void team(void);
int open(void);
void exercise1(void);
int error = 0;
void checkj(char);
void checkk(char);
void checkl(char);
void checksemicolon(char);
void checkf(char f);
void checkd(char d);
void checks(char s);
void checka(char a);
void checkq(char q);
void checkw(char w);
void checke(char e);
void checkr(char r);
void checkt(char t);
void checky(char y);
void checku(char u);
void checki(char o);
void checkp(char p);
void checkg(char h);
void checkz(char z);
void checkx(char x);
void checkc(char v);
```

```
void checkb(char b);
void checkn(char n);
void checkm(char m);
char input1(void);
char input2(void);
char input3(void);
char input4(void);
char input5(void);
char input6(void);
char input7(void);
char input8(void);
int main()

{   int opt,otpopen;
label:
opt= mainmenu();
while(1)
{switch (opt)
{
case 1:
  otpopen=open();
 if(otpopen==1)
 { exercise1();
 opt=5;
 goto label;
 }
 if(otpopen==2)
 { exercise1();}
 if(otpopen==3)
 {exercise1();
 }
 if(otpopen==4)
 {exercise1();
 }
  if(otpopen==5)
  { goto label;
   mainmenu();
   }
   case 4:
   CLRSCR;
  textcolor(LIGHTMAGENTA);
  cprintf("\nThis section is still under development");
    getch();
    goto label;
```

```
    case 2:
    textcolor(LIGHTMAGENTA);
    cprintf("\nThis section is still under development");
   getch();
   goto label;
  case 3:
   team();
   mainmenu();
   goto label;
case  5:
    return 0;
    break;

default:


    break;}
}

    return 0;
}
void draw(int opt, char *menu[], int n)
{
    int  i,bgcl[2] = {
        BLACK, LIGHTGRAY};
        box(28,10,25,n*2+2,YELLOW,BLACK);
        textcolor(14);
    cputs(menu[n]);

    for (i = 0; i < n; i++)
    {
        textbackground(bgcl[opt == (i + 1)]);
        cprintf(menu[i]);
    }
}
void box(int x, int y, int width, int height, int fc, int bc)
{
    int i;

    x -= 1;
    y -= 1;
    width += 2;
    height += 3;
    window(x, y, x + width, y + height);
```

```
    textbackground(bc);
    textcolor(fc);
    for (i = 2; i < width; i++)
   {     gotoxy(i, 1);
    putch(HB);
    gotoxy(i, height);
    putch(HB);
}
for (i = 2; i < height; i++)
{
    gotoxy(1, i);
    putch(VB);
    gotoxy(width, i);
    putch(VB);
}
gotoxy(width, 1);
putch(TRC);
gotoxy(1, height);
putch(BLC);
gotoxy(1, 1);
putch(TLC);
gotoxy(width, height);
putch(BRC);

x += 1;
y += 1;
width -= 2;
height -= 2;

window(x, y, x + width - 1, y + height - 1);
clrscr();
}
int mainmenu()
{
    char *menuitem[6] = {"\n\n\r (1) PLAY",
                         "\n\n\r (2) SET TIMER",
                         "\n\n\r (3) About Developer",
                         "\n\n\r (4) Your Progress",
                        " \n\n\r (5) Quit",
                         "  ***MAIN MENU***"};

    int c;
    int extended;
    int option = 1;
```

```
    CLRSCR
    textcolor(LIGHTGREEN);
    cprintf("\n\r%26s%c T Y P I N G  T U T O R %c", "", 16, 17);
    textcolor(LIGHTMAGENTA);
    cprintf("\n\n\r \xaf Press the arrow keys: \x18 or \x19 to navigate the options.\
\n\r \xaf Press <ENTER> key to accept an option.\
\n\r \xaf [OR] Press the number keys to select your choice");
    _setcursortype(_NOCURSOR);
    draw(1, menuitem, 5);
    while ((c = getch()) != '\r')
    {
        if (!c)
        {
            extended = getch();
            switch (extended)
            {
            case UP:
                option--;
                if (option < 1)
                    option = 5;
                break;
            case DOWN:
                option++;
                if (option > 5)
                    option = 1;
                break;
            }
        }
        else
        {
            option = c - 48;
            draw(option, menuitem, 5);
            break;
        }
        draw(option, menuitem, 5);
            }
            _setcursortype(_NORMALCURSOR);
            return option;

}
 int open (void)
{
    char *exer[6]={"\n\n\r (1)EXERCISE 1",
                   "\n\n\r (2)EXERCISE 2",
```

```
                    "\n\n\r (3)EXERCISE 3" ,
        "\n\n\r (4)EXERCISE 4" ,

                     "\n\n\r (5)BACK",
                     "\n  CHOOSE A EXERCISE" };
int c;
    int extended;
    int option = 1;
    CLRSCR
    textcolor(LIGHTGREEN);
    cprintf("\n\r%26s%c T Y P I N G  T U T O R %c", "", 16, 17);
    textcolor(LIGHTMAGENTA);
    cprintf("\n\n\r \xaf Press the arrow keys: \x18 or \x19 to navigate the options.\
\n\r \xaf Press <ENTER> key to accept an option.\
\n\r \xaf [OR] Press the number keys to select your choice");
    _setcursortype(_NOCURSOR);
    draw(1, exer,5);
    while ((c = getch()) != '\r')
    {
        if (!c)
        {
            extended = getch();
            switch (extended)
            {
            case UP:
                option--;
                if (option < 1)
                        option = 5;
                break;
            case DOWN:
                option++;
                      if (option > 5)
                   option = 1;
                break;
            }
        }
        else
        {
            option = c - 48;
               draw(option, exer, 5);
            break;
        }
            draw(option,exer, 5);
            }
```

```
            _setcursortype(_NORMALCURSOR);
            return option;

}
void team(void)
{   int i;
 clrscr();
     CLRSCR
    textcolor(LIGHTMAGENTA);
    cprintf("\n\n\rDeveloped By \n\rVEDANT NEMADE\n\rVASANTI KACHARE\n\rABDUL
WAASI\n\rPRANJAL JADHAV");

    getch();


        mainmenu();

    return;
}
void exercise1(void)
{  int i;
    char f, d, s, a, j, k, l, semicolon;
    CLRSCR
       clrscr();
      printf("f  d  j  k\n");
      f=input1();
      checkf(f);
       d=input2();
      checkd(d);
       j=input3();
      checkj(j);
       k=input4();
      checkk(k);

      printf("\nNo. of errors in this section =%d\n", error);
      error=0;
      printf("a  s  l  ;\n");
      a=input1();
      checka(a);
       s=input2();
      checks(s);
       l=input3();
      checkl(l);
       semicolon=input4();
```

```
checksemicolon(semicolon);

printf("\nNo. of errors in this section =%d\n", error);
error=0;
printf("f  d  s  a\n");
f=input1();
checkf(f);
 d=input2();
checkd(d);
 s=input3();
checks(s);
 a=input4();
checka(a);

printf("\nNo. of errors in this section =%d\n", error);
error=0;
printf("j  k  l  ;\n");
j=input1();
checkj(j);
 k=input2();
checkk(k);
 l=input3();
checkl(l);
 semicolon=input4();
checksemicolon(semicolon);

printf("\nNo. of errors in this section =%d\n", error);
 error=0;
printf("f  d  s  a  j  k  l  ;\n");
f=input1();
checkf(f);
 d=input2();
checkd(d);
 s=input3();
checks(s);
 a=input4();
checka(a);
j=input5();
checkj(j);
 k=input6();
checkk(k);
 l=input7();
checkl(l);
 semicolon=input8();
```

```
      checksemicolon(semicolon);
    printf("\nNo. of errors in this section =%d\n", error);

      getch();


      return ;
}
char input1(void)
{
      char letter;
      scanf("%c", &letter);
      printf("\033[A\033[3C");
      fflush(stdin);
      return letter;
}
char input2(void)
{
      char letter;
      scanf("%c", &letter);
      printf("\033[A\033[6C");
      fflush(stdin);
      return letter;
}
char input3(void)
{
      char letter;
      scanf("%c", &letter);
      printf("\033[A\033[9C");
      fflush(stdin);
      return letter;
}
char input4(void)
{
      char letter;
      scanf("%c", &letter);
      printf("\033[A\033[12C");
      fflush(stdin);
      return letter;
}
char input5(void)
{
      char letter;
      scanf("%c", &letter);
```

```
    printf("\033[A\033[15C");
    fflush(stdin);
    return letter;
}
char input6(void)
{
    char letter;
    scanf("%c", &letter);
    printf("\033[A\033[18C");
    fflush(stdin);
    return letter;
}
char input7(void)
{
    char letter;
    scanf("%c", &letter);
    printf("\033[A\033[21C");
    fflush(stdin);
    return letter;
}
char input8(void)
{
    char letter;
    scanf("%c", &letter);
    printf("\n");
    fflush(stdin);
    return letter;
}
void checkj(char j)
{
    if (j == 'j')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkk(char k)
{
    if (k == 'k')
    {
        error = error;
```

```
    }
    else
    {
        error = error + 1;
    }
}
void checkl(char l)
{
    if (l == 'l')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checksemicolon(char semicolon)
{
    if (semicolon == ';')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkf(char f)
{
    if (f == 'f')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkd(char d)
{
    if (d == 'd')
    {
        error = error;
```

```
    }
    else
    {
        error = error + 1;
    }
}
void checks(char s)
{
    if (s == 's')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checka(char a)
{
    if (a == 'a')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkg(char g)
{
    if (g == 'g')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkh(char h)
{
    if (h == 'h')
    {
        error = error;
```

```
    }
    else
    {
        error = error + 1;
    }
}
void checkq(char q)
{
    if (q == 'q')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkw(char w)
{
    if (w == 'w')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checke(char e)
{
    if (e == 'e')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkr(char r)
{
    if (r == 'r')
    {
        error = error;
```

```
    }
    else
    {
        error = error + 1;
    }
}
void checkt(char t)
{
    if (t == 't')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checky(char y)
{
    if (y == 'y')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checku(char u)
{
    if (u == 'u')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checki(char i)
{
    if (i == 'i')
    {
        error = error;
```

```
    }
    else
    {
        error = error + 1;
    }
}
void checko(char o)
{
    if (o == 'o')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkp(char p)
{
    if (p == 'p')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkz(char z)
{
    if (z == 'z')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkx(char x)
{
    if (x == 'x')
    {
        error = error;
```

```
    }
    else
    {
        error = error + 1;
    }
}
void checkc(char c)
{
    if (c == 'c')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkv(char v)
{
    if (v == 'v')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkb(char b)
{
    if (b == 'b')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
void checkn(char n)
{
    if (n == 'n')
    {
        error = error;
```

```
    }
    else
    {
        error = error + 1;
    }
}
void checkm(char m)
{
    if (m == 'm')
    {
        error = error;
    }
    else
    {
        error = error + 1;
    }
}
```