ABOUT.H

```
// about.h
#ifndef ABOUT_H
#define ABOUT_H
//Declare the Player_casino struct in the header file
#define RESET "\033[0m"
#define RED "\033[91m"
#define GREEN "\033[32m"
#define YELLOW "\033[33m"
#define BLUE "\033[34m"
#define CYAN "\033[36m"

struct Player_casino {
    char name[50];
    long int id;
    int credits;
};




void clear_input_buffer() {
    int ch;
    while ((ch = getchar()) != '\n' && ch != EOF);
}

void delay(int milliseconds) {
    Sleep(milliseconds);
}
void autosave(struct Player_casino* player) {
```

```
    FILE *file, *tempFile;
    char line[100];
    int id, credits;
    char name[50];
    int found = 0;

    file = fopen("player_data.csv", "r");
    tempFile = fopen("temp.csv", "w");

    if (!file || !tempFile) {
        printf("Error opening file.\n");
        return;
    }


    while (fgets(line, sizeof(line), file)) {
        sscanf(line, "%[^,],%d,%d", name, &id, &credits);


        if (id == player->id) {
            fprintf(tempFile, "%s,%d,%d\n", player->name, player-
>id, player->credits);
            found = 1;
        } else {

            fprintf(tempFile, "%s", line);
        }
    }


    if (!found) {
```

```
        fprintf(tempFile, "%s,%d,%d\n", player->name, player->id,
player->credits);
    }

    fclose(file);
    fclose(tempFile);


    remove("player_data.csv");
    rename("temp.csv", "player_data.csv");

    printf("\nCredits autosaved successfully.\n");
}


int getIntegerInput() {
    int number;
    while (1) {
        //printf("Enter an integer: ");

        // Check if scanf successfully reads an integer
        if (scanf("%d", &number) == 1) {
            clear_input_buffer(); // Clear any extra input
            return number;      // Return the valid integer input
        } else {
            printf("Invalid input. Please enter an integer: ");
            clear_input_buffer(); // Clear invalid input
        }
    }
}
// Function definition
```

```
void about() {
    // Print details about the game
    printf(YELLOW "Welcome to Codecrafters!\n" RESET );
    printf( GREEN "This game is designed to be visually
appealing while avoiding complex graphics libraries.\n");
    printf("Our goal is to help players understand core concepts
while enjoying the gameplay experience.\n");
    printf("One of the key advantages of our game is the ability to
save credits, allowing players to keep track of their progress and
achievements.\n");
    printf("\n");

    // Problem statement
    printf("Problem Statement:\n");
    printf("To create a visually appealing game without complex
graphics libraries to understand concepts.\n");
    printf("\n");

    // End credits with different colors for each member
    printf("Team Members:\n");
    printf(RED "1. Harsimran Singh - 2401030148\n" RESET);
    printf(CYAN "2. Mahi Gupta - 2401030135\n" RESET);
    printf(YELLOW "3. Shreysi Gupta - 2401030157\n"
RESET);
    printf(BLUE "4. Aryan Sharma - 2401030159\n" RESET);
    delay(2000);
    printf("\n");
    printf(RED"                          .\n");
    printf("                        .:+:\n");
    printf("                       :=+++=-\n");
    printf("                     :=+++++++=:\n");
```

```
    printf("                    .=++++++++++=\n");
    printf("                  .-+++++++++++++++.-\n");
    printf("                .-+*+**************++.-\n");
    printf("                ++**********************+: \n");
    printf("              .. **********************+. .\n");
    printf("             :=*: +**********************+ :*=: \n");
    printf("            .=***: +**********************+ -***=.
\n");
    printf("           =*****+ :**********************+
=*****=\n");
    printf("         .+********
.**********************:.+********. \n");
    printf("         .+********.**********************
:*********. \n");
    printf("         +********=.+**********************
:********= \n");
    printf("         +********* =**********************
+********- \n");
    printf("         =********* :**********************+-
*********- \n");
    printf("         -*********=.:=*****************=:
=*********- \n");
    printf("          **********+: .+*************.
:+**********: \n");
    printf("          ************+. -+********+-
.=************+. \n");
    printf("          ***************-..-+***+-..-*************+
\n");
    printf("          +++************+= .=*=. -
+************+*+ \n");
```

```
    printf("            -*:.**********:   :**********:.*=
\n");
    printf("             *:  :=*********= =*********+:  -*-
\n");
    printf("             *-    :========== +==========:    =* \n");
    printf("             **  ..  .========= +=========.  .  .+* \n");
    printf("             +*. .==.   +====== +======+.  .==  :** \n");
    printf("             -*- .=++-:  :=**** +**+=:  :=++=  :*+ \n");
    printf("             :**  -++++=:  :-** +*-:  -=+++=:  =*: \n");
    printf("             :**  .=++++++.   = -   :+*++++=.  ** \n");
    printf("             .+*-  :=++++=         .=++++=.  .** \n");
    printf("             .+*+.   :--:           :--:   .=** \n");
    printf("              =**+=                    .=+**= \n");
    printf("              -*****=                  -*****: \n");
    printf("              -******+-.             .-+******: \n");
    printf("              :+*******+-.          .-+*********: \n");
    printf("               -**********+-        -+**********- \n");
    printf("                .=+*********+-     :**********++. \n");
    printf("                  :==========+:.:+*********+=: \n");
    printf("                   :+**++++++*++*++++***+:. \n");
    printf("                     -++++++++++++++++++- \n");
    printf("                       =++++++++++++=. \n");
    printf("                        :=+++++++++=: \n");
    printf("                          .+++++++: \n");
    printf("                            -+++- \n");
    printf("                             .-. \n");
}

//MATHS ASYLUM
void vault1(int score,struct Player_casino* player);
void vault2(int score,struct Player_casino* player);
```

```
void vault3(int score,struct Player_casino* player);
void vault4(int score,struct Player_casino* player);
void vault5(int score,struct Player_casino* player);
void vault6(int score,struct Player_casino* player);
void vault7(int score,struct Player_casino* player);
void vault8(int score,struct Player_casino* player);
void vault9(int score,struct Player_casino* player);
void vault10(int score,struct Player_casino* player);
void vault11(int score,struct Player_casino* player);
void MATHS_ASYLUM(struct Player_casino* player);

void MATHS_ASYLUM(struct Player_casino* player)
{
    int intial_bet=0;
    printf("\033[1;33m");
    printf("\" M A T H S   A S Y L U M :  E S C A P E  T H E E Q U A T I O N S \"\n\n\n\n");
    printf("\033[1;32m");
    printf("I am you Chamberlane, CASPER\n\n");
    printf("Your task is to reach the topmost vault 10 by solving mathematical problems and open the secret box using your collected points\n");
    printf("On giving correct answer you will be awarded 30 credits\n");
    printf("On giving wrong answer, credits will be deducted and you will come to Vault 1\n");
    printf("if u surived till end , you get a huge tressure \n ");
    printf("ALL THE BEST %s\n\n\n\n",player->name);
    printf("\033[1;36m");
    printf("your current balance is %d \n",player->credits);
    autosave(player);
```

```
    int points=0;
    vault1(points,player);

}




void vault1(int score,struct Player_casino* player)
{
    srand(time(0));
    printf("\033[1;33m");
    printf("WELCOME TO VAULT 1\n");
    printf("\033[1;32m");
    printf("Your Riddle is - ");
    int answer;
    int vault_1_reward=30;
    int ch=rand()%10;
    char response[10];
    printf("\033[1;0m");
    switch(ch)
    {
    case 0:
        printf("Solve this BODMAS equation - 8+3x(12-4)/2\n");
        answer=getIntegerInput();
        if(answer==20)
        {
            score+=vault_1_reward;

            printf("\"Yes! You did it!\"\n");
```

```
        printf("Your score is %d points\n",score);
        player->credits+=vault_1_reward;
        autosave(player);

        fflush(stdin);

        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault2(score, player);
        }  else {

           printf("Thanks for playing since you quit 20%% of the
score will be deducted");
           player->credits-= (score*0.2);
        }
      }
      else
      {

        printf("\"No way! You lost!\"\n");
        printf("Your score is %d points\n",score);
        printf("Try Again\n\n");
        autosave(player);

        fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {

               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 1:
         printf("Solve this BODMAS equation -  5^2-(6x3)+(10-
6)\n");
         answer=getIntegerInput();
         if(answer==14)
         {
            score+=vault_1_reward;

            printf("\"Victory is ours!\"\n");

            printf("Your score is %d points\n",score);
            player->credits+=vault_1_reward;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault2(score, player);
            }  else {

               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         else
         {
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);

            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
```

```
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  case 2:
    printf("Solve this BODMAS equation - 45/5+12x(3-1) \n");
    answer=getIntegerInput();
    if(answer==33)
    {
        score+=vault_1_reward;
        printf("\"Unbelievable!\"\n");
        printf("Your score is %d points\n",score);
        player->credits+=vault_1_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);

        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault2(score, player);
        }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    else
    {
        printf("\"We were so close!\"\n");
        printf("Your score is %d points\n",score);
        printf("Try Again\n\n");
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);

        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  case 3:
```

```
        printf("Solve this BODMAS equation - (6+4)x(12/3)-5\n");
        answer=getIntegerInput();
        if(answer==35)
        {
            score+=vault_1_reward;
            printf("\"You won!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_1_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);

            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault2(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        else
        {
            printf("\"Gutted!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
```

```
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);

            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 4:
         printf("Solve this BODMAS equation - 8+(15/3)^2-9\n");
         answer=getIntegerInput();
         if(answer==24)
         {
            score+=vault_1_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_1_reward;
            autosave(player);

            fflush(stdin);
```

```
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);

        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault2(score, player);
        }  else {
           printf("Thanks for playing since you quit 20%% of the
score will be deducted");
           player->credits-= (score*0.2);
           autosave(player);
        }

      }
      else
      {
        printf("\"Oh no\"\n");
        printf("Your score is %d points\n",score);
        printf("Try Again\n\n");
        autosave(player);


        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
```

```
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  case 5:
    printf("Solve this BODMAS equation - 4x(7+2)-3^2\n");
    answer=getIntegerInput();
    if(answer==27)
    {
        score+=vault_1_reward;
        printf("\033[1;32m");
        printf("\"Can't believe you pulled it off!\"\n");
        printf("Your score is %d points\n",score);
        player->credits+=vault_1_reward;
        autosave(player);

        fflush(stdin);
        printf("\033[1;36m");
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault2(score, player);
        }  else {
```

```
            printf("\033[1;33m");
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }

    }
    else
    {   printf("\033[1;32m");
        printf("\"This is so frustrating!\"\n");
        printf("\033[1;33m");
        printf("Your score is %d points\n",score);
        printf("\033[1;32m");
        printf("Try Again\n\n");
        autosave(player);

        fflush(stdin);
        printf("\033[1;36m");
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("\033[1;33m");
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
    break;
  case 6:
    printf("Solve this BODMAS equation -
36/(2x3)+8x(2+1)\n");
    answer=getIntegerInput();
    if(answer==30)
    {
        score+=vault_1_reward;
        printf("\"Score!\"\n");
        printf("Your score is %d points\n",score);
        player->credits+=vault_1_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault2(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
```

```
        else
        {
            printf("\"Ugh, we almost had it!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    case 7:
        printf("Solve this BODMAS equation - (5+2x4)-
(3^2+6)\n");
        answer=getIntegerInput();
        if(answer==-2)
        {
            score+=vault_1_reward;
            printf("\"We nailed it!\"\n");
```

```
            printf("Your score is %d points\n",score);
            player->credits+=vault_1_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault2(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         else
         {
            printf("\"What a bummer!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 8:
         printf("Solve this BODMAS equation - 12+8/3-
(4+2)^2\n");
         answer=getIntegerInput();
         if(answer==-18)
         {
            score+=vault_1_reward;
            printf("\"Champion!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_1_reward;
            autosave(player);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault2(score, player);
            }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    else
    {
        printf("\"Defeated!\"\n");
        printf("Your score is %d points\n",score);
        printf("Try Again\n\n");
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  case 9:
```

```
        printf("Solve this BODMAS equation - (3x4)+9/3-
7+2^3\n");
        answer=getIntegerInput();
        if(answer==17)
        {
            score+=vault_1_reward;
            printf("\"Incredible!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_1_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault2(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted \n");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        else
        {
            printf("\"Not our day!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
```

```
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            };
         }
         break;
      default:
         printf("Solve this BODMAS equation - 20/4+6x(3+1)-
5\n");
         answer=getIntegerInput();
         if(answer==24)
         {
            score+=vault_1_reward;
            printf("\"Absolutely phenomenal!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_1_reward;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault2(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         else
         {
            printf("\"Next time, for sure!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
  }
}


void vault2(int score,struct Player_casino* player)
{
   srand(time(0));
   printf("\033[1;33m");
   printf("WELCOME TO VAULT 2\n");
   printf("\033[1;32m");
   printf("Your Riddle is - ");
   printf("\033[1;0m");
   int answer;
   float answer_7;
   int vault_2_reward = 30;
   int valut_2_loss=20;
   int ch=rand()%10;
   char response[10];
   switch(ch)
   {
   case 0:
      printf("Find the number of terms in an A.P. whose first
term a=5, common difference d=3, and last term l=50\n");
      answer=getIntegerInput();
      if(answer==16)
```

```
        {
            score+=vault_2_reward;
            printf("\"Yes! You did it!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+= vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault3(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score-=valut_2_loss;
            printf("\"No way! You lost!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-=valut_2_loss;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 1:
         printf("Find the 8th term of a G.P. where first term a=3 and
common ratio r=2\n");
         answer=getIntegerInput();
         if(answer==384)
         {
            score+=vault_2_reward;
            printf("\"Victory is ours!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault3(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score-=valut_2_loss;
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-=valut_2_loss;
            autosave(player);
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
```

```
            }
        }
        break;
    case 2:
        printf("If 12, x, and 4 are in H.P. , then find the value of
x\n");
        answer=getIntegerInput();
        if(answer==6)
        {
            score+=vault_2_reward;
            printf("\"Unbelievable!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault3(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
```

```
            score-=valut_2_loss;
            printf("\"You were so close!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-=valut_2_loss;
            autosave(player);
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 3:
         printf("If the sum of first 20 terms of an A.P. is 400 and
20th term is 50, then find the first term\n");
         answer=getIntegerInput();
         if(answer==10)
         {
            score+=vault_2_reward;
            printf("\"You won!\"\n");
            printf("Your score is %d points\n",score);
```

```
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault3(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score-=valut_2_loss;
            printf("\"Gutted!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-=valut_2_loss;
            autosave(player);
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    case 4:
        printf("In a GP, the 4th term is 16 and the 7th term is 128.
Find the common ratio\n");
        answer=getIntegerInput();
        if(answer==2)
        {
            score+=vault_2_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault3(score, player);
            }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    }
    else
    {
        score-=20;
        printf("\"Oh no, not again!\"\n");
        printf("Your score is %d points\n",score);
        printf("Try Again\n\n");
        player->credits-=valut_2_loss;
        autosave(player);
        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  case 5:
```

```
        printf("The first and second terms of an HP are 10 and 15,
respectively. Find the third term\n");
        answer=getIntegerInput();
        if(answer==30)
        {
            score+=vault_2_reward;
            printf("\"Can't believe we pulled it off!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault3(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score-=valut_2_loss;
            printf("\"This is so frustrating!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
```

```
            player->credits-=valut_2_loss;
            autosave(player);
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 6:
         printf("If the 6th term of an AP is 20 and the 10th term is
36, find the first term\n");
         answer=getIntegerInput();
         if(answer==0)
         {
            score+=vault_2_reward;
            printf("\"We nailed it!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault3(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score-=valut_2_loss;
            printf("\"Ugh, we almost had it!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-= valut_2_loss;
            autosave(player);
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  case 7:

    printf(" Find the sum to infinity of a GP where a=5 and
r=0.6\n");
    scanf("%f",&answer_7);
    if(answer_7==12.5)
    {
        score+=vault_2_reward;
        printf("\"Score!\"\n");
        printf("Your score is %d points\n",score);
        player->credits+=vault_2_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault3(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
```

```
            player->credits-= (score*0.2);
        }
    }
    else
    {
        score-=valut_2_loss;
        printf("\"What a bummer!\"\n");
        printf("Your score is %d points\n",score);
        printf("Try Again\n\n");
        player->credits-=valut_2_loss;
        autosave(player);
        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  case 8:
    printf("If 8, x, 24 are in H.P., then find the value of x\n");
    answer=getIntegerInput();
    if(answer==12)
```

```
        {
            score+=vault_2_reward;
            printf("\"Champion!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault3(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score-=valut_2_loss;
            printf("\"You got crushed!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-=valut_2_loss;
            autosave(player);
            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1? 
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response, 
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the 
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 9:
         printf("Find the number of terms in an AP where the first 
term a=3, the common difference d=5, and the last term 
l=78\n");
         answer=getIntegerInput();
         if(answer==16)
         {
            score+=vault_2_reward;
            printf("\"Incredible!\"\n");
            printf("Your score is %d points\n",score);
            player->credits+=vault_2_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault? 
(yes/YES/y): (not continuing will have consequences) ");
```

```
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault3(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score-=valut_2_loss;
            printf("\"Heartbreaking!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-=valut_2_loss;
            autosave(player);
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
    break;
  default:
    printf("The second term of a GP is 12, and the fifth term is
96. Find the common ratio\n");
    answer=getIntegerInput();
    if(answer==2)
    {
        score+=vault_2_reward;
        printf("\"Winner!\"\n");
        printf("Your score is %d points\n",score);
        player->credits+=vault_2_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault3(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    }
    else
```

```
        {
            score-= valut_2_loss;
            printf("\"Next time, for sure!\"\n");
            printf("Your score is %d points\n",score);
            printf("Try Again\n\n");
            player->credits-=valut_2_loss;
            autosave(player);
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
    }
}

void vault3(int score, struct Player_casino* player) {
    srand(time(0));
    printf("\033[1;33m");
    printf("WELCOME TO VAULT 3\n");
    printf("\033[1;32m");
    printf("Your Trigonometry Challenge is - ");
```

```
    printf("\033[1;0m");
    int vault_reward = 30, vault_loss = 20;
    int answer;
    float answer_7;
    int ch = rand() % 10;
    char response[10];

    switch (ch) {
    case 0:
        printf("What is the value of sin(30°) + cos(60°)?\n");
        answer=getIntegerInput();
        if (answer == 1) {
            score += vault_reward;
            printf("\"Yes! You did it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault4(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
```

```
        } else {
            score -= vault_loss;
            printf("\"No way! You lost!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 1:
        printf("If tan(x) = 1 and x is in the range [0°, 90°], what is
x?\n");
        answer=getIntegerInput();
        if (answer == 45) {
            score += vault_reward;
```

```
            printf("\"Victory is ours!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault4(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 2:
         printf("Calculate sin^2(45°) + cos^2(45°).\n");
         answer=getIntegerInput();
         if (answer == 1) {
            score += vault_reward;
            printf("\"Unbelievable!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault4(score, player);
```

```
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"You were so close!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 3:
```

```
        printf("What is the value of sec(60°)? (Round to nearest
whole number)\n");
        answer=getIntegerInput();
        if (answer == 2) {
            score += vault_reward;
            printf("\"You won!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault4(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"Gutted!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 4:
         printf("If cos(x) = 0.5 and x is between 0° and 90°, find
x.\n");
         answer=getIntegerInput();
         if (answer == 60) {
            score += vault_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
```

```
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault4(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         } else {
            score -= vault_loss;
            printf("\"Oh no, not again!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
```

```
            }
        }
        break;

    case 5:
        printf("Find the value of cot(45°).\n");
        answer=getIntegerInput();
        if (answer == 1) {
            score += vault_reward;
            printf("\"Can't believe we pulled it off!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault4(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"This is so frustrating!\"\n");
            printf("Your score is %d points\n", score);
```

```
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 6:
        printf("What is the value of sin(90° - 30°)?\n");
        scanf("%f", &answer_7);
        if (answer_7 == 0.5) {
            score += vault_reward;
            printf("\"We nailed it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault4(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         } else {
            score -= vault_loss;
            printf("\"Ugh, we almost had it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;

  case 7:
    printf("If sin(A) = 0.6 and A is acute, find cos(A) to the
nearest whole number.\n");
    scanf("%f", &answer_7);
    if (answer_7 == 0.8) {
        score += vault_reward;
        printf("\"Score!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault4(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
```

```
            }
        } else {
            score -= vault_loss;
            printf("\"What a bummer!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 8:
        printf("What is the value of sin(30°) * cos(60°)? (nearest
whole number)\n");
        scanf("%f", &answer_7);
        if (answer_7 == 0.25) {
```

```
            score += vault_reward;
            printf("\"Champion!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault4(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"You got crushed!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 9:
         printf("If tan(2A) = 1, where 0° < 2A < 90°, what is A?\n");
         scanf("%f", &answer_7);
         if (answer_7 == 22.5) {
            score += vault_reward;
            printf("\"Incredible!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault4(score, player);
```

```
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"Heartbreaking!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    default:
```

```
        printf("If sec(45°) = x, find x to the nearest whole
number.\n");
        scanf("%f", &answer_7);
        if (answer_7 == 1.414) {
            score += vault_reward;
            printf("\"Winner!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault4(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"Next time, for sure!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
      }
}


void vault4(int score, struct Player_casino* player)
{
   srand(time(0));
   printf("\033[1;33m");
   printf("WELCOME TO VAULT 4\n");
   printf("\033[1;32m");
   printf("Your Coordinate Geometry Challenge is - ");
   printf("\033[1;0m");
   int answer;
   float answer_7;

   int vault_reward = 30;
```

```
    int vault_loss = 20;
    char response[10];
    int ch = rand() % 10;
    switch (ch)
    {
    case 0:
        printf("Find the distance between the points (3, 4) and (7,
1). (Round to the nearest whole number)\n");
        answer=getIntegerInput();
        if (answer == 5)
        {
            score += vault_reward;
            printf("\"Yes! You did it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault5(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
```

```
        else
        {
            score -= vault_loss;
            printf("\"No way! You lost!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    case 1:
        printf("Find the midpoint of the line segment connecting (2,
-3) and (8, 5).\n");
        int x_mid, y_mid;
        scanf("%d %d", &x_mid, &y_mid);
        if (x_mid == 5 && y_mid == 1)
```

```
        {
            score += vault_reward;
            printf("\"Victory is ours!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault5(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 2:
         printf("Find the slope of a line passing through points (1, 2)
and (3, 6).\n");
         answer=getIntegerInput();
         if (answer == 2)
         {
            score += vault_reward;
            printf("\"Unbelievable!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault5(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score -= vault_loss;
            printf("\"You were so close!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
    break;
case 3:
    printf("What is the equation of a line with slope 2 passing through point (1, 3)? (Enter answer as ax+by+c=0, with integer coefficients)\n");
    int a, b, c;
    scanf("%d %d %d", &a, &b, &c);
    if (a == 2 && b == -1 && c == -1)
    {
        score += vault_reward;
        printf("\"You won!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault? (yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response, "YES") == 0 || strcmp(response, "y") == 0)
        {
            vault5(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the score will be deducted");
            player->credits-= (score*0.2);
        }
```

```
        }
        else
        {
            score -= vault_loss;
            printf("\"Gutted!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    case 4:
        printf("Find the radius of a circle with equation x^2 + y^2 -
4x + 6y - 12 = 0.\n");
        answer=getIntegerInput();
        if (answer == 5)
```

```
        {
            score += vault_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault5(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"Oh no, not again!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 5:
         printf("Find the length of the perpendicular drawn from
point (1, -2) to the line 3x + 4y + 5 = 0.\n");
         answer=getIntegerInput();
         if (answer == 5)
         {
            score += vault_reward;
            printf("\"Can't believe we pulled it off!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault5(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score -= vault_loss;
            printf("\"This is so frustrating!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
    break;
case 6:
    printf("Find the x-intercept of a line 2x - 3y + 6 = 0.\n");
    answer=getIntegerInput();
    if (answer == -3)
    {
        score += vault_reward;
        printf("\"We nailed it!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault5(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    }
    else
    {
```

```
            score -= vault_loss;
            printf("\"Ugh, we almost had it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 7:
         printf("If the slope of a line is -3 and it passes through (2,
5), what is the y-intercept?\n");
         answer=getIntegerInput();
         if (answer == 11)
         {
            score += vault_reward;
            printf("\"Score!\"\n");
```

```
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault5(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"What a bummer!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      case 8:
         printf("Find the area of a triangle with vertices at (0,0),
(6,0), and (3,6).\n");
         answer=getIntegerInput();
         if (answer == 18)
         {
            score += vault_reward;
            printf("\"Champion!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
```

```
            vault5(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    }
    else
    {
        score -= vault_loss;
        printf("\"You got crushed!\"\n");
        printf("Your score is %d points\n", score);
        printf("Try Again\n\n");
        player->credits -= vault_loss;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
```

```
        break;
    case 9:
        printf("Find the equation of a line passing through (3, -2)
and parallel to 4x - 5y = 7.\n");
        int par_a, par_b, par_c;
        scanf("%d %d %d", &par_a, &par_b, &par_c);
        if (par_a == 4 && par_b == -5 && par_c == -23)
        {
            score += vault_reward;
            printf("\"Incredible!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault5(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
```

```
            printf("\"Heartbreaking!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
      default:
         printf("What is the slope of a line perpendicular to one with
slope 3/4?\n");
         answer=getIntegerInput();
         if (answer == -4)
         {
            score += vault_reward;
            printf("\"Winner!\"\n");
            printf("Your score is %d points\n", score);
```

```
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault5(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"Next time, for sure!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    }
}


void vault5(int score, struct Player_casino* player)
{
    srand(time(0));
    printf("\033[1;33m");
    printf("WELCOME TO VAULT 5\n");
    printf("\033[1;32m");
    printf("Your Permutation and Combination Challenge is - ");
    printf("\033[1;0m");
    int answer;
    float answer_7; // for cases requiring float answers
    int vault_reward = 30; // reward points
    int vault_loss = 40; // loss points
    int ch = rand() % 10;
    char response[10];
    switch (ch)
```

```
    {
        case 0:
            printf("How many ways can you arrange the letters in the
word 'BOOK'? (Consider identical letters)\n");
            answer=getIntegerInput();
            if (answer == 12)
            {
                score += vault_reward;
                printf("\"Yes! You did it!\"\n");
                printf("Your score is %d points\n", score);
                player->credits += vault_reward;
                autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault6(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
            }
            else
            {
                score -= vault_loss;
                printf("\"No way! You lost!\"\n");
```

```
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
        }
        break;
      case 1:
        printf("In how many ways can you choose 3 students
from a class of 10?\n");
        answer=getIntegerInput();
        if (answer == 120)
        {
            score += vault_reward;
            printf("\"Victory is ours!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
```

```
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault6(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
        }
        else
        {
            score -= vault_loss;
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
            }
            break;
         case 2:
            printf("If a committee of 4 is to be formed from 8
people, how many different committees can be formed?\n");
            answer=getIntegerInput();
            if (answer == 70)
            {
               score += vault_reward;
               printf("\"Unbelievable!\"\n");
               printf("Your score is %d points\n", score);
               player->credits += vault_reward;
               autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault6(score, player);
            }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
          }
          }
          else
          {
            score -= vault_loss;
            printf("\"You were so close!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

          fflush(stdin);
          printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
          scanf("%s",response);
          if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
          {
            vault1(score, player);
          }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
          }
          }
          break;
        case 3:
```

```
        printf("How many different ways can 5 books be
arranged on a shelf?\n");
        answer=getIntegerInput();
        if (answer == 120)
        {
           score += vault_reward;
           printf("\"You won!\"\n");
           printf("Your score is %d points\n", score);
           player->credits += vault_reward;
           autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault6(score, player);
        }  else {
           printf("Thanks for playing since you quit 20%% of the
score will be deducted");
           player->credits-= (score*0.2);
        }
        }
        else
        {
           score -= vault_loss;
           printf("\"Gutted!\"\n");
           printf("Your score is %d points\n", score);
           printf("Try Again\n\n");
```

```
            player->credits -= vault_loss;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
        }
        break;
    case 4:
        printf("If you have 7 different colored balls, how many
ways can you choose 2 balls?\n");
        answer=getIntegerInput();
        if (answer == 21)
        {
            score += vault_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
```

```
        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault6(score, player);
        }  else {
           printf("Thanks for playing since you quit 20%% of the
score will be deducted");
           player->credits-= (score*0.2);
        }
        }
        else
        {
           score -= vault_loss;
           printf("\"Oh no, not again!\"\n");
           printf("Your score is %d points\n", score);
           printf("Try Again\n\n");
           player->credits -= vault_loss;
           autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault1(score, player);
```

```
        } else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
        }
        break;
      case 5:
        printf("How many different ways can the letters of the
word 'MISSISSIPPI' be arranged?\n");
        answer=getIntegerInput();
        if (answer == 34650)
        {
            score += vault_reward;
            printf("\"Can't believe we pulled it off!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault6(score, player);
        } else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
```

```
            player->credits-= (score*0.2);
        }
        }
        else
        {
            score -= vault_loss;
            printf("\"This is so frustrating!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
        }
        break;
    case 6:
        printf("In how many ways can you select 3 cards from a
standard deck of 52 cards?\n");
```

```
        answer=getIntegerInput();
        if (answer == 22100)
        {
            score += vault_reward;
            printf("\"We nailed it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault6(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
        }
        else
        {
            score -= vault_loss;
            printf("\"Ugh, we almost had it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
            }
            break;
         case 7:
            printf("If you roll a die, how many different outcomes
can you get?\n");
            answer=getIntegerInput();
            if (answer == 6)
            {
               score += vault_reward;
               printf("\"Score!\"\n");
               printf("Your score is %d points\n", score);
               player->credits += vault_reward;
               autosave(player);

            fflush(stdin);
```

```
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault6(score, player);
        }  else {
           printf("Thanks for playing since you quit 20%% of the
score will be deducted");
           player->credits-= (score*0.2);
        }
        }
        else
        {
           score -= vault_loss;
           printf("\"What a bummer!\"\n");
           printf("Your score is %d points\n", score);
           printf("Try Again\n\n");
           player->credits -= vault_loss;
           autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault1(score, player);
        }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
        }
        break;
      case 8:
        printf("How many ways can 4 students be seated in a
row of 4 chairs?\n");
        answer=getIntegerInput();
        if (answer == 24)
        {
            score += vault_reward;
            printf("\"Champion!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault6(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
```

```
            }
            }
            else
            {
               score -= vault_loss;
               printf("\"You got crushed!\"\n");
               printf("Your score is %d points\n", score);
               printf("Try Again\n\n");
               player->credits -= vault_loss;
               autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
            }
            break;
         case 9:
            printf("How many ways can you form a team of 3 from a
group of 6 people?\n");
            answer=getIntegerInput();;
```

```
        if (answer == 20)
        {
            score += vault_reward;
            printf("\"Incredible!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault6(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
        }
        else
        {
            score -= vault_loss;
            printf("\"Heartbreaking!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
            }
            break;
         default:
            printf("How many different ways can you arrange the
digits in the number 112233?\n");
            answer=getIntegerInput();
            if (answer == 90)
            {
               score += vault_reward;
               printf("\"Winner!\"\n");
               printf("Your score is %d points\n", score);
               player->credits += vault_reward;
               autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
```

```
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault6(score, player);
        }  else {
           printf("Thanks for playing since you quit 20%% of the
score will be deducted");
           player->credits-= (score*0.2);
        }
        }
        else
        {
           score -= vault_loss;
           printf("\"Next time, for sure!\"\n");
           printf("Your score is %d points\n", score);
           printf("Try Again\n\n");
           player->credits -= vault_loss;
           autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
           vault1(score, player);
        }  else {
           printf("Thanks for playing since you quit 20%% of the
score will be deducted");
```

```
            player->credits-= (score*0.2);
            autosave(player);
        }
        }
    }
}


void vault6(int score, struct Player_casino* player)
{
    srand(time(0));
    printf("\033[1;33m");
    printf("WELCOME TO VAULT 6\n current score- %d
\n",score);
    printf("\033[1;32m");
    printf("Your Limits and Continuity Challenge is - ");
    printf("\033[1;0m");

    float answer_7;
    int  vault_reward = 30, vault_loss = 40;
    int ch = rand() % 10;
    char response[10];
    switch (ch)
    {
    case 0:
        printf("What is the limit of f(x) = x^2 as x approaches
2?\n");
        scanf("%f", &answer_7);
        if (answer_7 == 4.0)
        {
            score += vault_reward;
```

```
            printf("\"Yes! You did it!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault7(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score -= vault_loss;
            printf("\"No way! You lost!\"\nYour score is %d
points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
```

```
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 1:
         printf("Evaluate the limit: lim (x -> 1) (x^2 - 1)/(x - 1).\n");
         scanf("%f", &answer_7);
         if (answer_7 == 2.0)
         {
            score += vault_reward;
            printf("\"Victory is ours!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
                vault7(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"I can't believe it!\"\nYour score is %d points\n",
score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
```

```
        }
        break;

    case 2:
        printf("If f(x) = 1/x, is f continuous at x = 0?\n(Answer 1
for Yes, 0 for No)\n");
        scanf("%f", &answer_7);
        if (answer_7 == 0.0)
        {
            score += vault_reward;
            printf("\"Unbelievable!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault7(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
```

```
            score -= vault_loss;
            printf("\"You were so close!\"\nYour score is %d
points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 3:
        printf("What is the limit of f(x) = (2x^2 - 8)/(x - 4) as x
approaches 4?\n");
        scanf("%f", &answer_7);
        if (answer_7 == 8.0)
        {
            score += vault_reward;
```

```
            printf("\"You won!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault7(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score -= vault_loss;
            printf("\"Gutted!\"\nYour score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 4:
         printf("Find the limit of f(x) = (x^2 - 1)/(x - 1) as x
approaches 1.\n");
         scanf("%f", &answer_7);
         if (answer_7 == 2.0)
         {
            score += vault_reward;
            printf("\"That was epic!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
                vault7(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"Oh no, not again!\"\nYour score is %d
points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
```

```
        }
        break;

    case 5:
        printf("What is the limit of f(x) = (3x^2 + 2x)/(x) as x
approaches 0?\n");
        scanf("%f", &answer_7);
        if (answer_7 == 0.0)
        {
            score += vault_reward;
            printf("\"Can't believe we pulled it off!\"\nYour score is
%d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault7(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
```

```
            score -= vault_loss;
            printf("\"This is so frustrating!\"\nYour score is %d
points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 6:
        printf("Is the function f(x) = |x| continuous at x = 0?
(Answer 1 for Yes, 0 for No)\n");
        scanf("%f", &answer_7);
        if (answer_7 == 1.0)
        {
            score += vault_reward;
```

```
            printf("\"We nailed it!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault7(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"Ugh, we almost had it!\"\nYour score is %d
points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
```

```
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 7:
         printf("What is the limit of f(x) = sin(x)/x as x approaches
0?\n");
         scanf("%f", &answer_7);
         if (answer_7 == 1.0)
         {
            score += vault_reward;
            printf("\"Score!\"\nYour score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
                vault7(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"What a bummer!\"\nYour score is %d points\n",
score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
```

```
        }
        break;

    case 8:
        printf("Evaluate the limit: lim (x -> ∞) (2x^2 -
3x)/(x^2).\n");
        scanf("%f", &answer_7);
        if (answer_7 == 2.0)
        {
            score += vault_reward;
            printf("\"Champion!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault7(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
```

```
            score -= vault_loss;
            printf("\"You got crushed!\"\nYour score is %d
points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 9:
        printf("Determine the continuity of f(x) = {x, x < 1; 1, x =
1; 2, x > 1} at x = 1. (Answer 1 for Continuous, 0 for
Discontinuous)\n");
        scanf("%f", &answer_7);
        if (answer_7 == 0.0)
        {
```

```
            score += vault_reward;
            printf("\"Incredible!\"\nYour score is %d points\n",
score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault7(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score -= vault_loss;
            printf("\"Heartbreaking!\"\nYour score is %d points\n",
score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      default:
         printf("Evaluate the limit: lim (x -> 3) (x^3 - 27)/(x -
3).\n");
         scanf("%f", &answer_7);
         if (answer_7 == 27.0)
         {
            score += vault_reward;
            printf("\"Winner!\"\nYour score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault7(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            score -= vault_loss;
            printf("\"Next time, for sure!\"\nYour score is %.0f
points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
}
```

```
void vault7(int score, struct Player_casino* player) {
    srand(time(0));
    printf("\033[1;33m");
    printf("WELCOME TO VAULT 7\n");
    printf("\033[1;32m");
    printf("Your Probability Challenge is - ");
    printf("\033[1;0m");
    char answer[20];  // Answer format: "x/y" or "x/y and z/w"
    int vault_reward = 25;   // Reward points for correct answers
    int vault_loss = 60;     // Points deducted for incorrect answers
    float variable_7;        // Placeholder for any float calculations
    int ch = rand() % 10;
    char response[10];
    switch (ch) {
    case 0:
        printf("If a six-sided die is rolled twice, what is the
probability of rolling a sum of 8 or more?\n");
        printf("Answer in the format 'x/y': ");
        scanf("%s", answer);
        if (strcmp(answer, "15/36") == 0) {
            score += vault_reward;
```

```
            printf("Correct! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault8(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         } else {
            score -= vault_loss;
            printf("Incorrect! You've lost %d points.\n", vault_loss);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
```

```
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;

  case 1:
    printf("In a group of 10 people, what is the probability that
at least two share the same birthday?\n");
    scanf("%s", answer);
    if (strcmp(answer, "11/12") == 0) {
        score += vault_reward;
        printf("Great! You've gained %d points!\n",
vault_reward);
        player->credits += vault_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault8(score, player);
        }  else {
```

```
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    } else {
        score -= vault_loss;
        printf("Sorry, that's wrong! You've lost %d points.\n",
vault_loss);
        player->credits -= vault_loss;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;

  case 2:
    printf("You draw 3 cards from a deck of 52 without
replacement. What is the probability that all are hearts?\n");
```

```
        scanf("%s", answer);
        if (strcmp(answer, "11/850") == 0) {
            score += vault_reward;
            printf("Well done! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault8(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("Incorrect! You've lost %d points.\n", vault_loss);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 3:
        printf("If you roll a die 3 times, what is the probability of
rolling exactly one 5?\n");
        scanf("%s", answer);
        if (strcmp(answer, "75/216") == 0) {
            score += vault_reward;
            printf("Nice job! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
```

```
            vault8(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    } else {
        score -= vault_loss;
        printf("Oh no! You've lost %d points.\n", vault_loss);
        player->credits -= vault_loss;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;

  case 4:
```

```
        printf("A box contains 6 red and 4 green balls. If you draw two balls without replacement, what is the probability that both are red?\n");
        scanf("%s", answer);
        if (strcmp(answer, "3/15") == 0) {
            score += vault_reward;
            printf("Excellent! You've gained %d points!\n", vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault? (yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response, "YES") == 0 || strcmp(response, "y") == 0)
            {
                vault8(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("Incorrect! You've lost %d points.\n", vault_loss);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 5:
        printf("If two coins are flipped, what is the probability that
both are heads?\n");
        scanf("%s", answer);
        if (strcmp(answer, "1/4") == 0) {
            score += vault_reward;
            printf("That's correct! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault8(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         } else {
            score -= vault_loss;
            printf("Incorrect! You've lost %d points.\n", vault_loss);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;
```

```
    case 6:
        printf("You roll a six-sided die and draw a card from a
standard deck. What is the probability of rolling a 4 and drawing
a Queen?\n");
        scanf("%s", answer);
        if (strcmp(answer, "1/78") == 0) {
            score += vault_reward;
            printf("Correct! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault8(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("Sorry, that's incorrect! You've lost %d points.\n",
vault_loss);
            player->credits -= vault_loss;
```

```
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 7:
        printf("What is the probability of drawing two aces
consecutively from a deck of 52 cards without replacement?\n");
        scanf("%s", answer);
        if (strcmp(answer, "1/221") == 0) {
            score += vault_reward;
            printf("Correct! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault8(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         } else {
            score -= vault_loss;
            printf("Incorrect! You've lost %d points.\n", vault_loss);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            fgets(answer, sizeof(answer), stdin);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
```

```
            }
        }
        break;

    case 8:
        printf("If a fair coin is flipped twice, what is the probability
of getting at least one tail?\n");
        clear_input_buffer();
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';

        if (strcmp(answer, "3/4") == 0) {
            score += vault_reward;
            printf("Great! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);


            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault8(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
```

```
        } else {
            score -= vault_loss;
            printf("Incorrect! You've lost %d points.\n", vault_loss);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 9:
        printf("A bag contains 5 black and 7 white marbles. If two
marbles are drawn without replacement, what is the probability
that both are black?\n");
        scanf("%s", answer);
        if (strcmp(answer, "10/66") == 0) {
            score += vault_reward;
```

```
            printf("Well done! You've gained %d points!\n",
vault_reward);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault8(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("Incorrect! You've lost %d points.\n", vault_loss);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
```

```
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;
  }
}


void vault8(int score, struct Player_casino* player)
{
  clear_input_buffer();
  printf("\033[1;33m");
  printf("\n current score %d \nWELCOME TO VAULT
8\n",score);
  printf("\033[1;32m");
  printf("Your Probability Challenge is - ");
  printf("\033[1;0m");
  char answer[50];  // To store answer as a string for
comparison
  int vault_reward = 30;
  int vault_loss = 70;
  int ch = rand() % 10;
  char response[10];
  vault_reward = 30;
  switch (ch)
  {
```

```
    case 0:
        printf("If two fair six-sided dice are rolled, what is the
probability that the sum of the numbers rolled is greater than 9?
What is the probability of rolling a sum of 7 given that the sum
is greater than 9?\n");
        printf("Answer in the format 'x/y and z/w': ");
        scanf("%[^\n]", answer);
        if (strcmp(answer, "5/36 and 1/3") == 0)
        {   vault_reward = 30;
            score += vault_reward;
            printf("\"Yes! You did it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault9(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
```

```
            vault_loss=20;
            score -= vault_loss;
            printf("\"No way! You lost!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 1:
        printf("In a standard deck of 52 cards, you draw 4 cards
without replacement. What is the probability that exactly 2 of
them are Aces and 2 are Kings?\n");
        scanf("%[^\n]", answer);
        if (strcmp(answer, "132/270725") == 0)
```

```
        {
            vault_reward = 30;
            score += vault_reward;
            printf("\"Victory is ours!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault9(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            vault_loss=20;
            score -= vault_loss;
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;


    case 2:
        printf("A box contains 5 red balls, 3 blue balls, and 2 green
balls. If two balls are drawn without replacement, what is the
probability that the second ball drawn is blue given that the first
ball drawn was red?\n");
        scanf("%[^\n]", answer);
        if (strcmp(answer, "3/9") == 0)
        {
            vault_reward = 30;
            score += vault_reward;
            printf("\"Unbelievable!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
```

```
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault9(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            vault_loss=20;
            score -= vault_loss;
            printf("\"You were so close!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 3:
         printf("If the probability of event A occurring is 0.3 and the
probability of event B occurring is 0.5, what is the probability
that at least one of the events occurs? Additionally, if A and B
are dependent events with a joint probability of 0.1, what is the
probability of both A and B occurring?\n");
         scanf("%[^\n]", answer);
         if (strcmp(answer, "0.55 and 0.1") == 0)
         {
            vault_reward = 30;
            score += vault_reward;
            printf("\"You won!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault9(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            vault_loss=20;
            score -= vault_loss;
            printf("\"Gutted!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
```

```
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 4:
        printf("In a class of 30 students, 18 are girls, and 10 boys
play sports. If you randomly select a student, what is the
probability that the student is a boy given that they are involved
in sports?\n");
        scanf("%s", answer);
        if (strcmp(answer, "10/30") == 0)
        {
            vault_reward = 30;
            score += vault_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
```

```
            vault9(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    }
    else
    {
        vault_loss=20;
        score -= vault_loss;
        printf("\"Oh no, not again!\"\n");
        printf("Your score is %d points\n", score);
        printf("Try Again\n\n");
        player->credits -= vault_loss;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault1(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
            autosave(player);
        }
```

```
        }
        break;

    case 5:
        printf("If you flip a fair coin 5 times, what is the probability
of getting at least 3 heads? Additionally, if you have already
obtained 2 heads, what is the probability of getting a total of 4
heads after the 5 flips?\n");
        scanf("%[^\n]", answer);
        if (strcmp(answer, "0.5 and 0.5") == 0)
        {
            vault_reward = 30;
            score += vault_reward;
            printf("\"Can't believe we pulled it off!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault9(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
```

```
        }
        else
        {
            vault_loss=20;
            score -= vault_loss;
            printf("\"This is so frustrating!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 6:
        printf("A jar contains 6 green, 4 yellow, 2 red, and 3 blue
marbles. If you draw two marbles at random, what is the
```

```
probability that both are of the same color? If the first marble
drawn is green, what is the probability that the second marble
drawn is yellow?\n");
        scanf("%[^\n]", answer);
        if (strcmp(answer, "27/66 and 4/11") == 0)
        {
            vault_reward = 30;
            score += vault_reward;
            printf("\"We nailed it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault9(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        }
        else
        {
            score -= vault_loss;
            printf("\"Ugh, we almost had it!\"\n");
```

```
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 7:
        printf("In a survey of 200 people, 120 liked chocolate, 80
liked vanilla, and 50 liked both. What is the probability that a
randomly selected person likes neither chocolate nor vanilla? If
you randomly select one person who likes chocolate, what is the
probability they also like vanilla?\n");
        scanf("%[^\n]", answer);
        if (strcmp(answer, "30/200 and 50/120") == 0)
        {
```

```
            vault_reward = 30;
            score += vault_reward;
            printf("\"Score!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault9(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            vault_loss=20;
            score -= vault_loss;
            printf("\"That stings!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 8:
        printf("You draw 3 cards from a standard deck of 52 cards
without replacement. What is the probability that at least one
card is a Queen? If at least one card drawn is a Queen, what is
the probability that exactly one of the drawn cards is a
Queen?\n");
        scanf("%[^\n]", answer);
        if (strcmp(answer, "1 - (48/52)(47/51)(46/50) and
(12/52)(40/51)(39/50)/P(at least one Queen)") == 0)
        {
            vault_reward = 30;
            printf("reward is %d",vault_reward);
            score += vault_reward;
            printf("\nAwesome!!!\n");
```

```
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            printf("Your balance %d\n",player->credits);
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault9(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         }
         else
         {
            vault_loss=20;
            score -= vault_loss;
            printf("\"Back to the drawing board!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 9:
         printf("If a box contains 10 bulbs, 4 of which are defective,
and you randomly choose 3 bulbs, what is the probability that all
3 are non-defective? If you know that at least one of the selected
bulbs is defective, what is the probability that exactly one of the
three bulbs is defective?\n");
         scanf("%[^\n]", answer);
         if (strcmp(answer, "C(6,3)/C(10,3) and
(C(4,1)*C(6,2)/C(10,3))") == 0)
         {
            vault_reward = 30;
            score += vault_reward;
            printf("\"Yes! That's how it's done!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
```

```
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault9(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        }
        else
        {   vault_loss=20;
            score -= vault_loss;
            printf("\"Oh no, try again!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    }
}


void vault9(int score, struct Player_casino* player) {
    clear_input_buffer();
    printf("\033[1;33m");
    printf("WELCOME TO VAULT 9\n");
    printf("\033[1;32m");
    printf("Your Integration Challenge is - ");
    printf("\033[1;0m");
    char answer[100];
    int vault_reward = 30; // Reward points for correct answers
    int vault_loss = 80;   // Points deducted for incorrect answers
    int ch = rand() % 10;
    char response[10];

    switch (ch) {
    case 0:
        printf("Evaluate the integral ∫(0 to 1) x^3 * (1 - x)^2
dx.\n");
```

```
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "1/60") == 0) {
            score += vault_reward;
            printf("\"Yes! You did it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault10(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"No way! You lost!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 1:
         printf("Compute integration (0 to infinity) x^3 * e^(-x)
dx.\n");
         printf("Answer: ");
         fgets(answer, sizeof(answer), stdin);
         answer[strcspn(answer, "\n")] = '\0';
         if (strcmp(answer, "6") == 0) {
            score += vault_reward;
            printf("\"Victory is ours!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
```

```
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault10(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         } else {
            score -= vault_loss;
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
    break;

case 2:
    printf("Evaluate integration (1 to infinity) 1/(x^3) dx.\n");
    printf("Answer: ");
    fgets(answer, sizeof(answer), stdin);
    answer[strcspn(answer, "\n")] = '\0';
    if (strcmp(answer, "1/2") == 0) {
        score += vault_reward;
        printf("\"Unbelievable!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault10(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
        }
    } else {
```

```
            score -= vault_loss;
            printf("\"You were so close!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
        }
        break;

    case 3:
        printf("Find the value of integration (0 to π/2) sin^2(x) dx.
(If your answer contains pi, write Pi)\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "Pi/4") == 0) {
            score += vault_reward;
```

```
            printf("\"You won!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault10(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"Gutted!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 4:
        printf("Evaluate the Beta function B(2, 3).\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "1/12") == 0) {
            score += vault_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault10(score, player);
```

```
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"Oh no, not again!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 5:
```

```
        printf("Find the integral integration (0 to 1) x^2 ln(x)
dx.\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "-1/9") == 0) {
            score += vault_reward;
            printf("\"Can't believe we pulled it off!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault10(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"This is so frustrating!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 6:
        printf("Evaluate the integral integration (0 to 1) (ln(x))^2
dx.\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "2") == 0) {
            score += vault_reward;
            printf("\"We nailed it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
```

```
            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault10(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
         } else {
            score -= vault_loss;
            printf("\"Ugh, we almost had it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
```

```
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;

  case 7:
    printf("Calculate the integral integration (0 to infinity) e^(-
x^2) dx. (If your answer contains pi, write Pi)\n");
    printf("Answer: ");
    fgets(answer, sizeof(answer), stdin);
    answer[strcspn(answer, "\n")] = '\0';
    if (strcmp(answer, "sqrt(Pi)/2") == 0) {
        score += vault_reward;
        printf("\"Score!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);

        fflush(stdin);
        printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
        scanf("%s",response);
        if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
        {
            vault10(score, player);
        }  else {
            printf("Thanks for playing since you quit 20%% of the
score will be deducted");
            player->credits-= (score*0.2);
```

```
            }
        } else {
            score -= vault_loss;
            printf("\"That stings!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 8:
        printf("Using the Gamma function, evaluate Γ(5).\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "24") == 0) {
```

```
            score += vault_reward;
            printf("\"Awesome!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault10(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"Back to the drawing board!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
```

```
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 9:
         printf("Evaluate integration (0 to π/2) cos(x)^3 dx.\n");
         printf("Answer: ");
         fgets(answer, sizeof(answer), stdin);
         answer[strcspn(answer, "\n")] = '\0';
         if (strcmp(answer, "2/3") == 0) {
            score += vault_reward;
            printf("\"Yes! That's how it's done!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the next vault?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
```

```
            {
                vault10(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
            }
        } else {
            score -= vault_loss;
            printf("\"Oh no, try again!\"\n");
            printf("Your score is %d points\n", score);
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    }
```

```
}

void vault10(int score, struct Player_casino* player)
{
    clear_input_buffer();
    printf("\033[1;33m");
    printf("WELCOME TO VAULT 10\n");
    printf("\033[1;32m");
    printf("Your Integration Challenge is - ");
    printf("\033[1;0m");
    char answer[100];
    int vault_reward = 30;
    int vault_loss = 90;
    int ch = rand() % 10;
    char response[10];
    switch (ch)
    {
    case 0:
        printf("Evaluate the double integral (0 to 1, 0 to 1) (x^2 +
y^2) dx dy.\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "1/3") == 0)
        {
            score += vault_reward;
            printf("\"Yes! You did it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
            vault11(score, player);
```

```
        }
        else
        {
            score -= vault_loss;
            printf("\"No way! You lost!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 1:
        printf("Compute double integral(0 to 1, 0 to sqrt(1 - x^2)) x
* y dy dx.\n");
        printf("Answer: ");
```

```
        fgets(answer, sizeof(answer), stdin);
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "1/8") == 0)
        {
            score += vault_reward;
            printf("\"Victory is ours!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
            vault11(score, player);
        }
        else
        {
            score -= vault_loss;
            printf("\"I can't believe it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
```

```
            player->credits-= (score*0.2);
            autosave(player);
        }
    }
    break;

  case 2:
    printf("Evaluate double integral(0 to 1, 0 to 1-x) (x + y) dx
dy.\n");
    printf("Answer: ");
    fgets(answer, sizeof(answer), stdin);
    answer[strcspn(answer, "\n")] = '\0';
    if (strcmp(answer, "1/3") == 0)
    {
        score += vault_reward;
        printf("\"Unbelievable!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);
        vault11(score, player);
    }
    else
    {
        score -= vault_loss;
        printf("\"You were so close!\"\n");
        printf("Your score is %d points\n", score);
        printf("Try Again\n\n");
        player->credits -= vault_loss;
        autosave(player);

        fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 3:
         printf("Find the value of f(x)(0 to 1, 0 to x) e^(x+y) dy
dx.\n");
         printf("Answer: ");
         fgets(answer, sizeof(answer), stdin);
         answer[strcspn(answer, "\n")] = '\0';
         if (strcmp(answer, "(e - 1)/2") == 0)
         {
            score += vault_reward;
            printf("\"You won!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
            vault11(score, player);
         }
```

```
        else
        {
            score -= vault_loss;
            printf("\"Gutted!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 4:
        printf("Evaluate the triple integral (0 to 1, 0 to x, 0 to y) z
dz dy dx.\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
```

```
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "1/24") == 0)
        {
            score += vault_reward;
            printf("\"That was epic!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
            vault11(score, player);
        }
        else
        {
            score -= vault_loss;
            printf("\"Oh no, not again!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
    break;

case 5:
    printf("Find the triple integral (0 to 1, 0 to x, 0 to y^2) z dx
dy dz.\n");
    printf("Answer: ");
    fgets(answer, sizeof(answer), stdin);
    answer[strcspn(answer, "\n")] = '\0';
    if (strcmp(answer, "1/12") == 0)
    {
        score += vault_reward;
        printf("\"Can't believe we pulled it off!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);
        vault11(score, player);
    }
    else
    {
        score -= vault_loss;
        printf("\"This is so frustrating!\"\n");
        printf("Your score is %d points\n", score);
        printf("Try Again\n\n");
        player->credits -= vault_loss;
        autosave(player);

        fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 6:
         printf("Evaluate the double integral (0 to π, 0 to π) sin(x) *
cos(y) dx dy.\n");
         printf("Answer: ");
         fgets(answer, sizeof(answer), stdin);
         answer[strcspn(answer, "\n")] = '\0';
         if (strcmp(answer, "0") == 0)
         {
            score += vault_reward;
            printf("\"We nailed it!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
            vault11(score, player);
         }
```

```
        else
        {
            score -= vault_loss;
            printf("\"Ugh, we almost had it!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;

    case 7:
        printf("Calculate the double integral (0 to 1, 0 to 1) (x^2 *
y + y^2 * x) dx dy.\n");
        printf("Answer: ");
        fgets(answer, sizeof(answer), stdin);
```

```
        answer[strcspn(answer, "\n")] = '\0';
        if (strcmp(answer, "11/120") == 0)
        {
            score += vault_reward;
            printf("\"Score!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
            vault11(score, player);
        }
        else
        {
            score -= vault_loss;
            printf("\"That stings!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
```

```
            autosave(player);
        }
    }
    break;

  case 8:
    printf("Using spherical coordinates, evaluate the triple
integral (B) sqrt(x^2 + y^2 + z^2) dV, where B is the unit
ball.\n");
    printf("Answer: ");
    fgets(answer, sizeof(answer), stdin);
    answer[strcspn(answer, "\n")] = '\0';
    if (strcmp(answer, "3/8") == 0)
    {
        score += vault_reward;
        printf("\"Awesome!\"\n");
        printf("Your score is %d points\n", score);
        player->credits += vault_reward;
        autosave(player);
        vault11(score, player);
    }
    else
    {
        score -= vault_loss;
        printf("\"Back to the drawing board!\"\n");
        printf("Your score is %d points\n", score);
        printf("Try Again\n\n");
        player->credits -= vault_loss;
        autosave(player);

        fflush(stdin);
```

```
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
               vault1(score, player);
            }  else {
               printf("Thanks for playing since you quit 20%% of the
score will be deducted");
               player->credits-= (score*0.2);
               autosave(player);
            }
         }
         break;

      case 9:
         printf("Evaluate the double integral (0 to 1, 0 to x) (x^3 -
y^3) dy dx.\n");
         printf("Answer: ");
         fgets(answer, sizeof(answer), stdin);
         answer[strcspn(answer, "\n")] = '\0';
         if (strcmp(answer, "1/8") == 0)
         {
            score += vault_reward;
            printf("\"Yes! That's how it's done!\"\n");
            printf("Your score is %d points\n", score);
            player->credits += vault_reward;
            autosave(player);
            vault11(score, player);
         }
```

```
        else
        {
            score -= vault_loss;
            printf("\"Oh no, try again!\"\n");
            printf("Your score is %d points\n", score);
            printf("Try Again\n\n");
            player->credits -= vault_loss;
            autosave(player);

            fflush(stdin);
            printf("Do you want to continue to the  vault 1?
(yes/YES/y): (not continuing will have consequences) ");
            scanf("%s",response);
            if (strcmp(response, "yes") == 0 || strcmp(response,
"YES") == 0 || strcmp(response, "y") == 0)
            {
                vault1(score, player);
            }  else {
                printf("Thanks for playing since you quit 20%% of the
score will be deducted");
                player->credits-= (score*0.2);
                autosave(player);
            }
        }
        break;
    }
}


void vault11(int score, struct Player_casino* player)
{
```

```
    printf("\033[1;32m");
    printf("congratulation %s !!!!!!!!!!! YOU WON!!
\n\n",player->name);
    printf("Since you survied such a dangreous journey your all
credits will be 100X\n\n");
    player->credits*=100;
    autosave(player);
    printf("\033[1;33m");
    printf("Your current Balance: %d ",player->credits);

}

#endif // ABOUT_H
```

GAME.C

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include <windows.h>
#include "about.h"

#define BOARD_SIZE 100
#define MAX_PLAYERS 4
#define DELAY 2000
#define DELAY_Card 1000
#define GRID_SIZE 3
#define MAX_DISTANCE 100
#define NUM_HORSES 7
#define SPEED_CHANGE_INTERVAL (MAX_DISTANCE /
100)
#define TOTAL_CARDS 52
#define MAX_PLAYERS_leaderboard 100
#define short_delay 800

typedef struct {
    char name[50];
    int position;
} Player;
```

```
typedef struct {
    char name[50];
    char symbol;
} Player_Ticktaktoe;
/*
struct Player_casino {
    char name[50];
    long int id;
    int credits;
};
*/
typedef struct {
    char suit;
    int rank;
} Card;

int snakes[5][2] = {{17, 7}, {54, 34}, {62, 19}, {64, 60}, {87,
24}};
int ladders[5][2] = {{3, 22}, {5, 8}, {11, 26}, {20, 29}, {27,
1}};

void clearConsole() {
    system("cls");
}

void gameMenu(struct Player_casino* player);
void snakeAndLadder();
void playTicTacToe(struct Player_casino* player_1);
void playrock_paper_scissor(struct Player_casino* player);
void displayMenu();
int rollDice();
```

```
int checkSnake(int position);
int checkLadder(int position);
void playSnakeAndLadder(Player players[], int numPlayers);
void printSeparator(int turnCount);
void delayPrint(int seconds);
void drawBoard(Player players[], int numPlayers,int turn_board
, int turn, int rolledNumbers[], int snakeFlag, int ladderFlag);

void playGuessingGame(struct Player_casino* player);
void casinoHub(struct Player_casino* player);
void loadPlayer(struct Player_casino* player);
void savePlayer(struct Player_casino* player);
void horseBettingGame(struct Player_casino* player);
void autosave(struct Player_casino* player);
void displayTrack(int positions[], int betHorse);
int randomSpeed();
void card_game(struct Player_casino* player);
void initializeDeck(Card deck[], int deckSize);
void shuffleDeck(Card deck[], int deckSize);
void displayCard(const Card* card);
void displayPlayerCards(const Card playerHand[], int
handSize);
Card drawCard(Card deck[], int* top);
int getPlayerChoice();
int getReplacementChoice();
void waitForPlayer();
void updateWinnerHand(Card winnerHand[], Card chosenCard,
int playedIndex);
void updateLoserHand(Card loserHand[], int playedIndex, Card
deck[], int* top);
int compareCredits(const void *a, const void *b);
```

```
void printTopPlayers(const char *filename);
void DICE_GAME(struct Player_casino* player);
void draw_dice(int dice_value);



void enableANSIColors() {
    HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
    DWORD dwMode = 0;
    GetConsoleMode(hOut, &dwMode);
    dwMode |=
ENABLE_VIRTUAL_TERMINAL_PROCESSING;
    SetConsoleMode(hOut, dwMode);
}

int main() {
    enableANSIColors();
    printf("\033[1;32m");
    printf(" .d8888b.         d8888 888b     d888
8888888888      888    888 888     888 888888b.   \n");
    printf("d88P  Y88b       d88888 8888b   d8888
888             888    888 888     888 888  \"88b  \n");
    printf("888    888      d88P888 88888b.d88888
888             888    888 888     888 888  .88P  \n");
    printf("888            d88P 888 888Y88888P888
8888888         8888888888 888     888 8888888K.  \n");
    printf("888  88888    d88P  888 888 Y888P 888
888             888    888 888     888 888  \"Y88b \n");
    printf("888    888   d88P   888 888  Y8P  888
888             888    888 888     888 888    888 \n");
```

```
    printf("Y88b  d88P  d8888888888 888   \"   888
888            888    888 Y88b. .d88P 888   d88P \n");
    printf(" \"Y8888P88 d88P     888 888       888
8888888888     888    888  \"Y88888P\"  8888888P\"   \n\n\n\n\
n");
    printf("\033[1;31m");

    printf("\n**********************\n");
    printf("\033[1;93m");
    printf("      GAME HUB      \n");
    printf("  By - CodeCrafters  \n");
    printf("\033[1;31m");
    printf("**********************\n");
    printf("\033[1;36m");
    printf("Welcome!!! \n");
    struct Player_casino player;
    loadPlayer(&player);
    gameMenu(&player);
    return 0;
}

void gameMenu(struct Player_casino* player) {
    int option;

    do {
        displayMenu();
        printf("Choose an option: ");
        scanf("%d", &option);
        switch (option) {
            case 1:
                snakeAndLadder();
```

```
            break;
        case 2:
            srand(time(0));
            playTicTacToe(player);
            break;
        case 3:
            playrock_paper_scissor(player);
            break;
        case 4:
            casinoHub(player);
            break;
        case 7:
            printf("Exiting Game Hub. Goodbye!\n");
            exit(0);
        case 5:
            MATHS_ASYLUM(player);
            break;
        case 6:
            about(player);
            break;
        default:
            printf("Invalid option, please try again.\n");
      }
   } while (option != 4);
}

void displayMenu() {
   printf("\033[1;31m");
   printf("\n======= MENU =======\n");
   printf("\033[1;93m");
   printf("1. Snake and Ladder\n");
```

```
    printf("2. Tic-Tac-Toe\n");
    printf("3. rock paper scissor\n");
    printf("4. Credits Hub\n");
    printf("5. MATHS ASYLUM \n");
    printf("6. About\n");
    printf("7. EXIT\n");
    printf("\033[1;31m");
    printf("====================\n");
    printf("\033[1;36m");
}

//code for Snake and ladder



void delayPrint(int seconds) {

    Sleep(seconds * 1000);

}


// Draw the game board with player positions
void drawBoard(Player players[], int numPlayers,int turn_board
, int turn, int rolledNumbers[], int snakeFlag, int ladderFlag) {
    clearConsole();

    // Set up the board as a 10x10 grid
    int board[10][10];
    int value = 100;
```

```
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            if (i % 2 == 0) {
                board[i][j] = value--;
            } else {
                board[i][9 - j] = value--;
            }
        }
    }

    printf("Snake and Ladder Game Board: Turn %d\n",
turn_board);
    printf("-----------------------------------------------------------------
------------------------------\n");

    // Show rolled numbers
    printf("\033[1;96m");
    printf("Rolled Numbers:\n");
    for (int i = 0; i < numPlayers; i++) {
        printf("\033[1;%dm",31+i);
        printf("%s rolled: %d\n", players[i].name,
rolledNumbers[i]);
    }
    printf("\033[1;96m");
    printf("\n");

    // Show player positions
    printf("Player Positions:\n");
    for (int i = 0; i < numPlayers; i++) {
        printf("\033[1;%dm",31+i);
```

```
        printf("%s is at Tile %d\n", players[i].name,
players[i].position);
    }
    printf("\n");

    if (snakeFlag) {
        printf("\033[1;31m");
        printf("%s encountered a snake! Sliding down.\n",
players[turn].name);
    }
    if (ladderFlag) {
        printf("\033[1;32m");
        printf("%s found a ladder! Climbing up to %d.\n",
players[turn].name,players[turn].position);
    }
    printf("\033[38;2;255;165;0m");
    printf("----------------------------------------------------------------
----------------------------------------\n");

    // Display the 10x10 board with player positions
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            int tileNum = board[i][j];
            char playerChar[10] = "  ";

            for (int k = 0; k < numPlayers; k++) {
                if (players[k].position == tileNum) {

                    const char *colorCodes[] = {"\033[1;31m",
"\033[1;32m", "\033[1;33m", "\033[1;34m"};
```

```
        // Ensure k is within bounds of colorCodes array
                int colorIndex = k % 4; // Use modulo in case of
more than 4 players

        // Format player position with unique color
                snprintf(playerChar, sizeof(playerChar),
"%sP%d\033[0m", colorCodes[colorIndex], k + 1);
            }
          }
          printf("\033[38;2;255;165;0m");
          printf("| %5d %s ", tileNum, playerChar);
        }
        printf("|\n");

    }
    printf("-----------------------------------------------------------------
-----------------------------------------\n");
    printf("\033[1;32m");
    // Display snakes and ladders
    for (int i = 0; i < 5; i++) {
        printf("LS %d: %d to %d  | ", i + 1, ladders[i][0],
ladders[i][1]);
        printf("SH %d: %d to %d\n", i + 1, snakes[i][0],
snakes[i][1]);
    }
    printf("\nLS: Ladder start  SH: Snake Head\n");
    printf("\033[1;35m");
    printf("-------------------------------------------------------\n");
    printf("\033[1;0m");
}
```

```
void snakeAndLadder() {
    printf("\033[1;36m");
    int numPlayers;
    srand(time(NULL));

    printf("Enter the number of players (2 to 4): ");
    numPlayers = getIntegerInput();


    if (numPlayers < 2 || numPlayers > MAX_PLAYERS) {
        printf("Invalid number of players. Exiting...\n");
        return;
    }

    Player players[numPlayers];
    int rolledNumbers[numPlayers];
    for (int i = 0; i < numPlayers; i++) {
        rolledNumbers[i] = 0;
    }

    for (int i = 0; i < numPlayers; i++) {
        clear_input_buffer();
        printf("Enter name for Player(Type AI to play against AI)
%d: ", i + 1);
        scanf(" %[^\n]", players[i].name);

        players[i].position = 0;
    }

    int turn = 0;
    int turn_board = 0;
```

```
    char input[10];
    while (1) {
        printf("\033[1;35m");
        int roll = (rand() % 6) + 1;
        int snakeFlag = 0;
        int ladderFlag = 0;

        // Check if the player is AI
        if (strcmp(players[turn].name, "AI") != 0) {

            printf("%s's turn. Press Enter to roll the dice or type '404'
to quit: ", players[turn].name);
            fgets(input, sizeof(input), stdin);  // Read the input

            // Remove the newline character from input
            input[strcspn(input, "\n")] = 0;

            // Check if the player wants to quit
            if (strcmp(input, "404") == 0) {
                printf("Game terminated by %s.\n",
players[turn].name);
                break;
        }
        } else {
            if (turn_board % 5 == 0) {
            printf("%s's turn. Press Enter to roll the dice or type '404'
to quit: ", players[turn].name);
            fgets(input, sizeof(input), stdin);  // Read the input

            // Remove the newline character from input
            input[strcspn(input, "\n")] = 0;
```

```
            // Check if the player wants to quit
            if (strcmp(input, "404") == 0) {
                printf("Game terminated by %s.\n",
players[turn].name);
                break;
            }
            }
            printf("AI rolled automatically.\n");
            delay(short_delay-300);
        }

        players[turn].position += roll;
        rolledNumbers[turn] = roll;

        // Check for snake or ladder
        for (int j = 0; j < 5; j++) {
            if (players[turn].position == snakes[j][0]) {
                players[turn].position = snakes[j][1];
                snakeFlag = 1;
            } else if (players[turn].position == ladders[j][0]) {
                players[turn].position = ladders[j][1];
                ladderFlag = 1;
            }
        }

        if (players[turn].position > BOARD_SIZE) {
            players[turn].position = BOARD_SIZE;
        }
        turn_board++;
```

```
        drawBoard(players, numPlayers,turn_board,turn,
rolledNumbers, snakeFlag, ladderFlag);
        delay(DELAY);

        if (players[turn].position == BOARD_SIZE) {
            printf("%s wins!\n", players[turn].name);
            break;
        }

        turn = (turn + 1) % numPlayers;
    }
    printf("game over");
    printf("\033[1;0m");
}

// code for tick-tak-toe

void displayGrid(char grid[GRID_SIZE][GRID_SIZE]) {
    printf("\033[1;35m");  // Purple board title
    printf("Current Tic-Tac-Toe Board:\n");
    printf("  1 2 3\n");
    for (int i = 0; i < GRID_SIZE; i++) {
        printf("%d ", i + 1);
        for (int j = 0; j < GRID_SIZE; j++) {
            // Use different colors based on the symbol
            if (grid[i][j] == 'X') {
                printf("\033[1;32m");  // Green for X
            } else if (grid[i][j] == 'O') {
                printf("\033[1;31m");  // Red for O
            } else {
                printf("\033[0m");  // Reset color for empty spaces
```

```
            }
            printf("%c ", grid[i][j]);
            printf("\033[1;35m");  // Reset to purple for board layout
after each character
        }
        printf("\n");
    }
    printf("\033[0m");  // Reset all colors after the grid is
displayed
}

int checkWin(char grid[GRID_SIZE][GRID_SIZE], char
symbol) {
    for (int i = 0; i < GRID_SIZE; i++) {
        if (grid[i][0] == symbol && grid[i][1] == symbol &&
grid[i][2] == symbol)
            return 1;
        if (grid[0][i] == symbol && grid[1][i] == symbol &&
grid[2][i] == symbol)
            return 1;
    }
    if (grid[0][0] == symbol && grid[1][1] == symbol &&
grid[2][2] == symbol)
        return 1;
    if (grid[0][2] == symbol && grid[1][1] == symbol &&
grid[2][0] == symbol)
        return 1;
    return 0;
}

int isFull(char grid[GRID_SIZE][GRID_SIZE]) {
```

```
    for (int i = 0; i < GRID_SIZE; i++) {
        for (int j = 0; j < GRID_SIZE; j++) {
            if (grid[i][j] == ' ')
                return 0;
        }
    }
    return 1;
}

int canWin(char grid[GRID_SIZE][GRID_SIZE], char symbol,
int *winRow, int *winCol) {
    for (int i = 0; i < GRID_SIZE; i++) {
        for (int j = 0; j < GRID_SIZE; j++) {
            if (grid[i][j] == ' ') {
                grid[i][j] = symbol;
                if (checkWin(grid, symbol)) {
                    grid[i][j] = ' ';
                    *winRow = i;
                    *winCol = j;
                    return 1;
                }
                grid[i][j] = ' ';
            }
        }
    }
    return 0;
}

// AI move logic: smarter approach
void aiMove(char grid[GRID_SIZE][GRID_SIZE]) {
    int row, col;
```

```
int winRow, winCol;


if (canWin(grid, 'O', &winRow, &winCol)) {
    grid[winRow][winCol] = 'O';
    return;
}


if (canWin(grid, 'X', &winRow, &winCol)) {
    grid[winRow][winCol] = 'O';
    return;
}


if (grid[1][1] == ' ') {
    grid[1][1] = 'O';
    return;
}


int corners[4][2] = {{0, 0}, {0, 2}, {2, 0}, {2, 2}};
for (int i = 0; i < 4; i++) {
    if (grid[corners[i][0]][corners[i][1]] == ' ') {
        grid[corners[i][0]][corners[i][1]] = 'O';
        return;
    }
}


do {
```

```
        row = rand() % GRID_SIZE;
        col = rand() % GRID_SIZE;
    } while (grid[row][col] != ' ');
    grid[row][col] = 'O';
}

void playTicTacToe(struct Player_casino* player_1) {
    Player_Ticktaktoe player;
    char grid[GRID_SIZE][GRID_SIZE] = {{' ', ' ', ' '}, {' ', ' ', ' '},
{' ', ' ', ' '}};
    int currentPlayer;
    int row, col;
    char playAgainstAI='y';

    printf("Playing against AI: ");
    //scanf(" %c", &playAgainstAI);
    delay(DELAY);
    if (playAgainstAI == 'y' || playAgainstAI == 'Y') {


        memcpy(player.name,player_1->name,strlen(player_1-
>name)+1);
        //printf("\033[1;35m");
        player.symbol = 'X';
        currentPlayer = 0;
    }

    clearConsole();
    displayGrid(grid);

    while (1) {
```

```
        printf("\033[1;36m");
        if (currentPlayer == 0) {
            printf("\n%s, enter your move\n ", player.name);
            printf("enter row: ");
            row=getIntegerInput();
            printf("enter collumn: ");
            col=getIntegerInput();
            row--;
            col--;

            if (row < 0 || row >= GRID_SIZE || col < 0 || col >=
GRID_SIZE || grid[row][col] != ' ' ) {
                printf("Invalid move. Try again.\n");
                continue;
            }

            grid[row][col] = player.symbol;
        } else {
            aiMove(grid);
            printf("\nAI makes a move.\n");
        }

        clearConsole();
        displayGrid(grid);

        if (checkWin(grid, currentPlayer == 0 ? player.symbol :
'O')) {
            if (currentPlayer == 0) {
                printf("\033[1;36m");
                printf("\033[1;32m");
```

```
            printf("Congratulations %s! You win!\n 500 creidts
added in your account : ", player.name);
            player_1->credits+=500;
            printf("current balance - %d\n",player_1->credits);
            autosave(player_1);

        } else {
            printf("\033[1;31m");
            printf("AI wins! Better luck next time, %s.\n",
player.name);
        }
        break;
    } else if (isFull(grid)) {
        printf("\033[1;36m");
        printf("It's a draw!\n");
        break;
    }

    currentPlayer = (currentPlayer + 1) % 2;
  }
  printf("\033[1;36m");
}

const char* options[] = {"rock", "paper", "scissor"};

void quit_game(struct Player_casino* player) {
  printf("\033[1;32m");
  printf("Thanks for playing!\n");
  printf("\033[1;0m");
  gameMenu(player);
}
```

```
const char* turn() {
    int x = rand() % 3;
    return options[x];
}

void aftermath(struct Player_casino* player);
void new_game(struct Player_casino* player) {
    printf("\033[1;36m");
    int player_score = 0;
    int computer_score = 0;
    int turns, i;

    printf("%s  How many times would you like to play? ",player-
>name);
    turns=getIntegerInput();
    if (turns > 10) {
        printf("too many rounds!!!:\n rounds set to 10:\n");
        turns=10;
    }

    for (i = 0; i < turns; i++) {
        printf("\033[1;31m");
        printf("########################################\n");

        char player_turn[10];
        const char* computer_turn;

        do {
            printf("\033[1;36m");
```

```
            printf("What do you want to draw: rock, paper, or
scissor? ");
            scanf("%s", player_turn);
            if (strcmp(player_turn, "rock") != 0 &&
strcmp(player_turn, "paper") != 0 && strcmp(player_turn,
"scissor") != 0) {
                printf("\033[1;31m");
                printf("Enter a valid input!\n");
            }
        } while (strcmp(player_turn, "rock") != 0 &&
strcmp(player_turn, "paper") != 0 && strcmp(player_turn,
"scissor") != 0);
        printf("\033[1;31m");
        printf("#######################################\n\n")
;
        printf("\033[1;35m");
        printf("-----------------------------------\n");
        printf("\033[1;33m");
        printf("Your prompt is: %s\n", player_turn);
        computer_turn = turn();
        printf("Computer's prompt is: %s\n", computer_turn);
        printf("\033[1;35m");
        printf("-----------------------------------\n");

        if (strcmp(player_turn, computer_turn) == 0) {
            printf("\033[1;32m");
            printf("Oh, a draw... try again.\n");
        } else if ((strcmp(player_turn, "rock") == 0 &&
strcmp(computer_turn, "scissor") == 0) ||
                   (strcmp(player_turn, "scissor") == 0 &&
strcmp(computer_turn, "paper") == 0) ||
```

```
                (strcmp(player_turn, "paper") == 0 &&
strcmp(computer_turn, "rock") == 0)) {
            printf("\033[1;32m");
            printf("Player won!\n");
            player_score++;
        } else {
            printf("\033[1;31m");
            printf("Computer won!\n");
            computer_score++;
        }
        printf("\033[1;37m");
        printf("Current score:\nPlayer score: %d\nComputer score:
%d\n", player_score, computer_score);
    }

    if (player_score > computer_score) {
        printf("\033[1;31m");
        printf("########################################\n");
        printf("\033[1;32m");
        printf("Congratulations %s, you win!\n %d credits added in
you acount\n",player->name,50*player_score);
        player->credits += 50*player_score;
        autosave(player);
        printf("\033[1;31m");
        printf("########################################\n");
    } else if (player_score == computer_score) {
        printf("\033[1;32m");
        printf("Ohhh, a draw! you get 10 credits\n");
        player->credits += 10;
        autosave(player);
    } else {
```

```
        printf("\033[1;31m");
        printf("########################################\n");
        printf("\033[1;32m");
        printf("Oops, you lose!\n");
        printf("\033[1;31m");
        printf("########################################\n");
    }
    printf("\033[1;0m");
    aftermath(player);
}

void aftermath(struct Player_casino* player) {
    char response[4];
    do {
        printf("\033[1;31m");
        printf("########################################\n");
        printf("\033[1;33m");
        printf("Wanna try again? (yes or no): ");
        scanf("%s", response);
        printf("\033[1;361");
        printf("########################################\n");
    } while (strcmp(response, "yes") != 0 && strcmp(response,
"no") != 0);

    if (strcmp(response, "yes") == 0) {
        new_game(player);
    } else {
        quit_game(player);
    }
}
```

```
void playrock_paper_scissor(struct Player_casino* player) {
    clearConsole();
    srand(time(0)); // Seed for random number generation
    char response[4];
    printf("\033[1;31m");
    printf("########################################\n");
    printf("\033[1;33m");
    printf("Welcome to Rock Paper Scissor\n");
    printf("\033[1;31m");
    printf("########################################\n");

    do {
        printf("\033[1;36m");
        printf("Wanna start the game? (yes or no): ");
        scanf("%s", response);
        printf("\033[1;31m");
        printf("########################################\n");
    } while (strcmp(response, "yes") != 0 && strcmp(response,
"no") != 0);

    if (strcmp(response, "yes") == 0) {
        new_game(player);
    } else {
        printf("\033[1;32m");
        printf("Bye bye!\n");
    }

    printf("\033[1;0m");

}
```

```
void loadPlayer(struct Player_casino* player) {
    FILE* file;
    char line[100];
    int found = 0;

    printf("Enter your name: ");
    scanf("%[^\n]", player->name);
    printf("Enter your unique ID: ");
    player->id=getIntegerInput();


    file = fopen("player_data.csv", "r");
    if (file == NULL) {

        file = fopen("player_data.csv", "w");

        player->credits = 500;
        fprintf(file, "%s,%ld,%d\n", player->name, player->id,
player->credits);
        printf("New player created! You have %d credits.\n",
player->credits);
        fclose(file);
        return;
    }


    while (fgets(line, sizeof(line), file)) {
        char name[50];
        int id, credits;
        sscanf(line, "%[^,],%ld,%d", name, &id, &credits);
        if (id == player->id) {
```

```
            strcpy(player->name, name);
            player->credits = credits;
            found = 1;
            printf("Welcome back, %s! You have %d credits.\n",
player->name, player->credits);
            break;
        }
    }

    if (!found) {

        fclose(file);
        file = fopen("player_data.csv", "a");
        player->credits = 500;
        fprintf(file, "%s,%ld,%d\n", player->name, player->id,
player->credits);
        printf("New player created! You have %d credits.\n",
player->credits);
    }

    fclose(file);
}

void savePlayer(struct Player_casino* player) {
    FILE* file, *tempFile;
    char line[100];
    int found = 0;

    file = fopen("player_data.csv", "r");
    tempFile = fopen("temp.csv", "w");
```

```
    //fprintf(tempFile, "Name,ID,Credits\n");

    while (fgets(line, sizeof(line), file)) {
        char name[50];
        int id, credits;
        sscanf(line, "%[^,],%ld,%d", name, &id, &credits);

        if (id == player->id) {
            fprintf(tempFile, "%s,%ld,%d\n", player->name, player-
>id, player->credits);
            found = 1;
        } else {
            fprintf(tempFile, "%s", line);
        }
    }

    fclose(file);
    fclose(tempFile);

    remove("player_data.csv");
    rename("temp.csv", "player_data.csv");
}

void casinoHub(struct Player_casino* player) {
    int choice;

    while(1) {
        printf("\033[1;31m");
        printf("\n--- Welcome to the Credits Hub, %s ---\n", player-
>name);
        printf("\033[1;32m");
```

```
        printf("WELCOME TO A GAME OF LUCK WHERE
YOUR CREDITS MATTERS!!!!\n\n\n");
        printf("\033[1;33m");
        printf("1. Play Guessing Game\n");
        printf("2. Horse betting game\n");
        printf("3. High Stakes Showdown\n");
        printf("4. Dice Rolling Simulator Game\n");
        printf("5. LeaderBoard\n");
        printf("6. Exit\n");
        printf("\033[1;36m");
        printf("Choose an option: ");

        choice =  getIntegerInput();

        switch (choice) {
            case 1:
                playGuessingGame(player);
                break;
            case 2:
                horseBettingGame(player);
                break;
            case 3:
                card_game(player);
            case 6:

                printf("Exiting the casino hub. Goodbye!\n");
                savePlayer(player);
                gameMenu(player);
                break;
            case 5:
                printTopPlayers("player_data.csv");
```

```
            break;
        case 4:
            DICE_GAME(player);
            break;
        default:
        printf("\031[1;36m");
        printf("Invalid option. Please try again.\n");
        printf("\033[1;36m");
      }
   }
}

void playGuessingGame(struct Player_casino* player) {
   int aiNumber, userGuess, attempts = 3, bet;
   time_t currentTime;
   char buffer[26];
   struct tm* tm_info;

   time(&currentTime);
   tm_info = localtime(&currentTime);
   strftime(buffer, 26, "%Y-%m-%d %H:%M:%S", tm_info);

   printf("\033[1;31m");
   printf("\n--- Guessing Game ---\n");
   printf("\033[1;32m");
   printf("AI has thought of a number between 1 and 10. If you
guess it within 3 rounds, your bet will be doubled. If not, you'll
lose 50%% of your bet.\n\n");

   printf("\033[1;33m");
   printf("You currently have %d credits.\n\n", player->credits);
```

```
    printf("\033[1;36m");
    printf("How much would you like to bet? ");
    bet=getIntegerInput();

    if (bet > player->credits) {
        printf("\033[1;31m");
        printf("You don't have enough credits to place that bet.\n");
        return;
    }

    srand(time(0));
    aiNumber = rand() % 10 + 1;


    for (int i = 0; i < attempts; i++) {
        printf("\033[1;35m");
        printf("Attempt %d: Enter your guess (1-10): ", i + 1);
        while(1) {
            userGuess=getIntegerInput();
            if (userGuess <= 10 && userGuess >=1) {break;}
            printf("\033[1;31m");
            printf("enter a valid number(1-10): ");
        }

        if (userGuess == aiNumber) {
            printf("\033[1;32m");
            printf("Congratulations! You've guessed the
number!\n");
            player->credits += bet * 2;
            printf("\033[1;33m");
```

```
            printf("Your new balance is %d credits.\n", player->credits);
            savePlayer(player);
            return;
        } else if (userGuess < aiNumber) {
            printf("\033[1;31m");
            printf("The AI's number is higher.\n");
        } else {
            printf("\033[1;32m");
            printf("The AI's number is lower.\n");
        }
    }

    printf("\033[1;37m");
    printf("Sorry, you've run out of attempts. The AI's number was %d.\n", aiNumber);
    player->credits -= bet / 2;
    printf("\033[1;33m");
    printf("Your new balance is %d credits.\n", player->credits);
    savePlayer(player);
    printf("\033[1;0m");
}


void displayTrack(int positions[], int betHorse) {
    clearConsole();
    printf("\033[1;31m");
    printf("=== Horse Betting Game ===\n");
    printf("\033[1;32m");
    printf("Track Length: %d\n", MAX_DISTANCE);
```

```
    // Define colors for each horse
    char* horseColors[NUM_HORSES] = {
        "\033[1;34m", // Blue
        "\033[1;35m", // Magenta
        "\033[1;36m", // Cyan
        "\033[1;33m", // Yellow
        "\033[1;37m"  // White
    };

    for (int i = 0; i < NUM_HORSES; i++) {
        printf("\033[1;32m"); // Reset to green for horse number
        printf("Horse %d: ", i + 1);

        for (int j = 0; j < MAX_DISTANCE; j++) {
            if (j == positions[i]) {
                // Print the horse in its designated color
                printf("%s\033[0m", horseColors[i % 5]);
            } else {
                // Print track dashes in orange color
                printf("\033[38;5;214m-\033[0m");
            }
        }

        // Mark the betting horse
        if (i + 1 == betHorse) {
            printf(" <-- Your Bet");
        }

        printf("\n");
    }
}
```

```
int randomSpeed() {
    //srand(time(0));
    return rand() % 10 + 1;
}

void horseBettingGame(struct Player_casino* player) {
    srand(time(0));
    int bet, betHorse;
    int positions[NUM_HORSES] = {0};
    int speeds[NUM_HORSES] = {0};

    printf("Welcome to the Horse Betting Game, %s!\n", player-
>name);
    printf("You have %d credits.\n", player->credits);
    printf("Enter your bet amount: ");
    bet = getIntegerInput();

    if (bet > player->credits) {
        printf("You don't have enough credits!\n");
        return;
    }

    printf("Select the horse you want to bet on (1-%d): ",
NUM_HORSES);
    while(1) {
        betHorse = getIntegerInput();
        if (betHorse  <= 7 && betHorse >= 1) {break;}
        printf("enter a valid horse number(1-7): ");
    }
```

```
srand(time(0));

for (int i = 0; i < NUM_HORSES; i++) {
    speeds[i] = randomSpeed();
    if (i==betHorse-1 && bet==404)  {
        speeds[i]+=10;
    }
    if (i==betHorse-1 && bet<=500)  {
        speeds[i]+=8;
    }
    if (i==betHorse-1 && bet>=1500)  {
        speeds[i]-=4;
    }

}

int finished = 0, winner = -1;

while (!finished) {
    printf("current balance: %d \n",player->credits);
    //delay(DELAY);
    for (int i = 0; i < NUM_HORSES; i++) {
        positions[i] += speeds[i];

        if (positions[i] >= MAX_DISTANCE) {
            positions[i] = MAX_DISTANCE;
            winner = i + 1;
            finished = 1;
        }
```

```
            if (positions[i] % SPEED_CHANGE_INTERVAL == 0)
{
                speeds[i] = randomSpeed();
                if (i==betHorse-1 && bet==404)  {
                    speeds[i]+=5;
                }
                if (i==betHorse-1 && bet==404)  {
                speeds[i]+=10;
                }
                if (i==betHorse-1 && bet<=500 && i>=50)  {
                    speeds[i]+=2;
                }
                if (i==betHorse-1 && bet>=1500  && i >= 50)  {
                    speeds[i]-=2;
                }

            }
        }

        displayTrack(positions, betHorse);
        usleep(500000);
    }
    printf("Horse %d wins the race!\n", winner);
    printf("current balance: %d \n",player->credits);

    if (winner == betHorse) {
        printf("Congratulations! Your horse won! win amount %d
* 10\n",bet);
        player->credits = (bet * 10 ) + player->credits;
    } else {
```

```
        printf("Sorry, you lost the bet %d.\n",bet);
        player->credits -= (bet);

    }

    printf("Your new balance is %d credits.\n", player->credits);
    autosave(player);
}

/*
void autosave(struct Player_casino* player) {
    FILE *file, *tempFile;
    char line[100];
    int id, credits;
    char name[50];
    int found = 0;

    file = fopen("player_data.csv", "r");
    tempFile = fopen("temp.csv", "w");

    if (!file || !tempFile) {
        printf("Error opening file.\n");
        return;
    }


    while (fgets(line, sizeof(line), file)) {
        sscanf(line, "%[^,],%d,%d", name, &id, &credits);


        if (id == player->id) {
```

```
            fprintf(tempFile, "%s,%d,%d\n", player->name, player->id, player->credits);
            found = 1;
        } else {

            fprintf(tempFile, "%s", line);
        }
    }


    if (!found) {
        fprintf(tempFile, "%s,%d,%d\n", player->name, player->id, player->credits);
    }

    fclose(file);
    fclose(tempFile);


    remove("player_data.csv");
    rename("temp.csv", "player_data.csv");

    printf("Credits autosaved successfully.\n");
}

*/

void card_game(struct Player_casino* player) {
    fflush(stdin);
    printf("\033[1;36m");
    printf("do you want know how to play this game?(Y/N) : ");
```

```
    char know;
    scanf("%c",&know);
    if(know=='Y'||know=='y')
    {
        printf("\033[1;32m");
        printf("\n                                                    HOW
TO PLAY\n");
        printf("In High Stakes Showdown, you and the computer
face off in a strategic battle of wits and luck. Start each round by
placing a bet with your credits, then choose one card from your
hand of three to play. The player with the higher card wins the
round and claims the pot. The winner can choose to keep their
played card or replace it with a new one, while the loser receives
a new random card. The game continues until one player runs
out of credits so stay sharp and play smart to emerge
victorious!\n\n\n");
        printf("\nS=Spade C=Club D=Diamond H=Hearts\n");
    }


    if (player->credits < 100) {
        printf("\033[1;31m");
        printf("insufficent Balance\n");
        exit(0);
    }
    Card deck[TOTAL_CARDS];
    Card playerHand[3], computerHand[3];
    int top = 0, playerCredits = 0, computerCredits = 0;
    int round = 1;
    while (1) {
        printf("\033[1;36m");
```

```
        printf("enter your bet for the game: ");
        playerCredits = getIntegerInput();

        if (playerCredits < player->credits && playerCredits >=
100) {
            break;
        }
        printf("\033[1;31m");
        printf("\n enter a valid bet(must be greater than OR equal
to  100)\n");
    }
    player->credits -= playerCredits;
    autosave(player);

    computerCredits=playerCredits;
    srand(time(0));



    initializeDeck(deck, TOTAL_CARDS);
    shuffleDeck(deck, TOTAL_CARDS);



    for (int i = 0; i < 3; i++) {
        playerHand[i] = drawCard(deck, &top);
        computerHand[i] = drawCard(deck, &top);
    }
    int current_bet=0;

    while ((playerCredits > 0 && computerCredits > 0 )|| (round
> 15)) {
        printf("\033[1;35m");
```

```
        printf("\n--- Round %d ---\n", round);
        printf("Player Credits: %d | Computer Credits: %d\n",
playerCredits, computerCredits);

        printf("\033[1;33m");
        printf("Your Hand:\n");
        printf("\033[1;0m");
        displayPlayerCards(playerHand, 3);
        while (1) {
            printf("\033[1;36m");
            printf("enter you current bet for this round: ");
            current_bet = getIntegerInput();
            if ((current_bet>computerCredits ||
current_bet>playerCredits) || (current_bet < 10)) {
                printf("\033[1;31m");
                printf("\nenter a valid bet!!! such that both players can
participate \n");
            } else {break;}
        }
        int playerChoice = getPlayerChoice() - 1;


        int computerChoice = rand() % 3;
        printf("\033[1;35m");
        printf("\nYou play: ");
        displayCard(&playerHand[playerChoice]);
        printf("\nComputer plays: ");
        displayCard(&computerHand[computerChoice]);


        Card playerCard = playerHand[playerChoice];
```

```
        Card computerCard = computerHand[computerChoice];
        Card newCard = drawCard(deck, &top);
        int playerWins = playerCard.rank > computerCard.rank;

        if (playerWins) {
            printf("\033[1;32m");
            printf("You win this round!\n");
            playerCredits += current_bet;
            computerCredits -= current_bet;

            autosave(player);

            printf("\033[1;32m");
            printf("\nChoose a card to keep:\n1. Your played
card\n2. Computer's played card\n3. New drawn card\n");
            printf("\033[1;0m");
            displayCard(&playerCard);
            displayCard(&computerCard);
            displayCard(&newCard);
            printf("\033[1;36m");

            int replacementChoice = getReplacementChoice();
            Card chosenCard = (replacementChoice == 1) ?
playerCard :
                              (replacementChoice == 2) ? computerCard :
newCard;


            updateWinnerHand(playerHand, chosenCard,
playerChoice);
```

```
            updateLoserHand(computerHand, computerChoice,
deck, &top);

        } else {
            printf("\033[1;35m");
            printf("Computer wins this round!\n");
            playerCredits -= current_bet;
            computerCredits += current_bet;

            autosave(player);

            int replacementChoice = rand() % 3;
            Card chosenCard = (replacementChoice == 0) ?
playerCard :
                              (replacementChoice == 1) ? computerCard :
newCard;


            updateWinnerHand(computerHand, chosenCard,
computerChoice);
            updateLoserHand(playerHand, playerChoice, deck,
&top);
        }
        delay(DELAY_Card);
        round++;
        waitForPlayer();
    }


    if (playerCredits <= 0 ) {
        printf("\033[1;32m");
```

```
        printf("\n\nGame Over! Computer Wins!\n");
    } else {
        printf("\033[1;32m");
        printf("\n\nCongratulations! You Win!\n");
        player->credits+=playerCredits;
        autosave(player);
    }

    if (round > 10) {
        printf("\033[1;31m");
        printf("maximum rounds reached:\n");
        if (playerCredits>computerCredits) {
            printf("\033[1;32m");
            printf("\n\nCongratulation %s you win!!\n",player-
>name);
                    player->credits+=playerCredits;
                    autosave(player);
        } else {
            printf("\033[1;32m");
            printf("\n\nGame Over! Computer Wins!\n");
                    player->credits+=playerCredits;
                    autosave(player);

        }
    }
    printf("\033[1;33m");
    printf("your current Balance is %d\n",player->credits);
    casinoHub(player);
    printf("\033[1;30m");
}
```

```
void initializeDeck(Card deck[], int deckSize) {
    char suits[] = {'H', 'D', 'S', 'C'};
    int cardIndex = 0;
    for (int i = 0; i < 4; i++) {
        for (int j = 1; j <= 13; j++) {
            deck[cardIndex].suit = suits[i];
            deck[cardIndex].rank = j;
            cardIndex++;
        }
    }
}

void shuffleDeck(Card deck[], int deckSize) {
    for (int i = deckSize - 1; i > 0; i--) {
        int j = rand() % (i + 1);
        Card temp = deck[i];
        deck[i] = deck[j];
        deck[j] = temp;
    }
}

void displayCard(const Card* card) {
    printf("\n|----------|\n");
    printf("| %c %2d     |\n", card->suit, card->rank);
    printf("|----------|\n\n");
}

void displayPlayerCards(const Card playerHand[], int handSize)
{
    for (int i = 0; i < handSize; i++) {
        printf("Card %d: ", i + 1);
```

```
        displayCard(&playerHand[i]);
    }
}

Card drawCard(Card deck[], int* top) {
    return deck[(*top)++];
}

int getPlayerChoice() {
    int choice;
    printf("Choose a card to play (1, 2, or 3): ");
    scanf("%d", &choice);
    while (choice < 1 || choice > 3) {
        printf("Invalid choice. Choose a card to play (1, 2, or 3): ");
        scanf("%d", &choice);
    }
    return choice;
}

int getReplacementChoice() {
    int choice;
    printf("Choose a replacement card (1, 2, or 3): ");
    scanf("%d", &choice);
    while (choice < 1 || choice > 3) {
        printf("Invalid choice. Choose a replacement card (1, 2, or
3): ");
        scanf("%d", &choice);
    }
    return choice;
}
```

```
void waitForPlayer() {
    printf("Press Enter to continue...");
    while (getchar() != '\n');
}

void updateWinnerHand(Card winnerHand[], Card chosenCard,
int playedIndex) {
    winnerHand[playedIndex] = chosenCard;
}

void updateLoserHand(Card loserHand[], int playedIndex, Card
deck[], int* top) {
    loserHand[playedIndex] = drawCard(deck, top);
}

int compareCredits(const void *a, const void *b) {
    struct Player_casino *playerA = (struct Player_casino *)a;
    struct Player_casino *playerB = (struct Player_casino *)b;
    return playerB->credits - playerA->credits;
}

// Function to load, sort, and print the top 10 players by credits
void printTopPlayers(const char *filename) {
    FILE *file = fopen(filename, "r");
    if (!file) {
        printf("Error opening file.\n");
        return;
    }

    struct Player_casino players[MAX_PLAYERS_leaderboard];
    int playerCount = 0;
```

```
    char line[100];

    // Skip the header line if there's one
    fgets(line, sizeof(line), file);

    // Read and parse the CSV data
    while (fgets(line, sizeof(line), file) && playerCount <
MAX_PLAYERS_leaderboard) {
        sscanf(line, "%[^,],%d,%d", players[playerCount].name,
&players[playerCount].id, &players[playerCount].credits);
        playerCount++;
    }
    fclose(file);

    // Sort the players by credits in descending order
    qsort(players, playerCount, sizeof(struct Player_casino),
compareCredits);
    printf("\033[1;0m");
    // Print the top 10 players
    printf("Top Players (by Credits):\n");
    printf("| Rank | Name                         | ID              |
Credits      |\n");
    printf("|------|------------------------------|-----------------|--------
------|\n");

    for (int i = 0; i < playerCount && i < 10; i++) {
    if (i==0) {
        printf("\033[1;33m");
        //
```

```
        printf("| %-4d | %-20s    \033[1;31m#1   \033[1;33m |
**********       | %-7d      |\n", i + 1,
players[i].name,  players[i].credits);
    } else {
        printf("\033[1;0m");
        printf("| %-4d | %-20s         | **********       | %-
7d      |\n", i + 1, players[i].name,  players[i].credits);
    }
    }

}

void DICE_GAME(struct Player_casino* player) {
    clear_input_buffer();
    printf("\033[1;36m");
    printf("do you want know how to play this game?(Y/N) : ");
    char know;
    scanf("%c",&know);
    if(know=='Y'||know=='y')
    {

        printf("\033[1;35m");
        printf("HOW TO PLAY : The goal is to roll two dice to get
a high score and 'lock' the score as your final choice.The player
and AI both get 5 turns each to roll the dice and decide whether
to lock or roll ahead the dice in each turn. After locking the user
will not get any further chances to roll the dice. The player with
the highest locked score wins the game.\n\n");
        delay(5000);

    }
```

```
char choice1='r',choice2='r';
int dice1,dice2,lock=0,lock2=0,flock=12,i=10;

// Seed the random number generator
srand(time(0));
printf("\033[1;33m");
printf("Welcome to the Dice Rolling Simulator!\n");

// Game loop
while(i>0)
{
   if(i%2==0)
   // Roll the dice (random number between 1 and 6)
   {
     if (choice1=='r'||choice1=='R')
     {
       system("cls");
       printf("\033[1;32m");
       printf("its %s turn!\n",player->name);
       dice1 = (rand() % 6) + 1;
       dice2 = (rand() % 6) + 1;
   // Clear the terminal screen
       delay(DELAY);
        // For Windows

   // Display the result
      printf("\033[1;33m");
      printf("You rolled a %d!\n", dice1+dice2);
      printf("\033[1;37m");
      draw_dice(dice1);
      draw_dice(dice2);
```

```
   printf("\033[1;36m");

 // Ask the user if they want to roll again
   printf("Press 'r' to roll again or 'l' to lock: ");
   scanf(" %c", &choice1);
   if(choice1=='l'||choice1=='L'||i==2)
   {
     lock=dice1+dice2;
     flock=dice1+dice2;
   }
 }
}

 else

 {

   if(choice2=='r'||choice2=='R')
   {
    system("cls");

    printf("it's AI's turn!");
    dice1 = (rand() % 6) + 1;
    dice2 = (rand() % 6) + 1;
 // Clear the terminal screen
    delay(DELAY);
     // For Windows

 // Display the result
   printf("\033[1;33m");
   printf("AI rolled a %d!\n", dice1+dice2);
```

```
      draw_dice(dice1);
      draw_dice(dice2);
      delay(2000);

    // saving AI's decision
      if(dice1+dice2>=10||dice1+dice2>flock)
        choice2='l';
       else
          choice2='r';
        if(choice2=='l')

          {
          printf("\033[1;32m");
          printf("\nAI HAS LOCKED IT'S CHOICE!\n");
          delay(2000);}

        //locking the score
         lock2=dice1+dice2;
      }
    }
    i--;

}
printf("\033[1;32m");
printf("\nSorry! no more chance is left!\n");
printf("your choice is : %d\n",lock);
printf("AI's choice is : %d\n",lock2);
if(lock==lock2) {
    printf("\033[1;35m");
    printf("IT'S A TIE! You won 100 credits \n");
    player->credits += 100;
```

```
        autosave(player);
    }
    else if(lock>lock2) {
        printf("\033[1;32m");
        printf("\nHURRAY! USER WINS! you won 500
credits\n");
        player->credits += 500;
        autosave(player);
    }
    else {
        printf("\033[1;32m");
        printf("\nAI WINS!BETTER LUCK NEXT TIME! you
lost 300 credits\n");
        player->credits-=300;
        if (player->credits<=0) {
            player->credits=1;
        }
        autosave(player);
    }
    printf("\033[1;33m");
    printf("\nThank you for playing! current Balance %d
\n",player->credits);

    printf("\033[1;0m");

}

// Function to display the dice as ASCII art
void draw_dice(int dice_value) {
    // Display the dice value as ASCII art
    switch (dice_value) {
```

```
case 1:
    printf("---------\n");
    printf("|       |\n");
    printf("|   o   |\n");
    printf("|       |\n");
    printf("---------\n");
    break;
case 2:
    printf("---------\n");
    printf("| o     |\n");
    printf("|       |\n");
    printf("|     o |\n");
    printf("---------\n");
    break;
case 3:
    printf("---------\n");
    printf("| o     |\n");
    printf("|   o   |\n");
    printf("|     o |\n");
    printf("---------\n");
    break;
case 4:
    printf("---------\n");
    printf("| o   o |\n");
    printf("|       |\n");
    printf("| o   o |\n");
    printf("---------\n");
    break;
case 5:
    printf("---------\n");
    printf("| o   o |\n");
```

```
            printf("|   o   |\n");
            printf("| o   o |\n");
            printf("---------\n");
            break;
        case 6:
            printf("---------\n");
            printf("| o   o |\n");
            printf("| o   o |\n");
            printf("| o   o |\n");
            printf("---------\n");
            break;
        default:
        printf("\033[1;31m");
            printf("Invalid dice value!\n");
            break;
    }
}
```