```c
#ifndef GFILES_H_INCLUDED
#define GFILES_H_INCLUDED

/* LIBRARIES UTILISED*/
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#include<windows.h>

/* CONSTANTS */
#define MAXYEAR 5    /// Max number of years to be played
#define MAXTURN 3       /// Max number of turns in a year
#define MINGOLD 1000    /// Min investment in Gold
#define MINRE 10000     /// Min investment in Real Estate
#define MINSTOCK 3000   /// Min investment in Stock Market

/* GLOBAL VARS */
int nPlays=0;           /// Number of time the game has been played
char playerList[10][40];     /// Matrix of player names
int scores[10];              /// Scores of the list of players

/* UDFs */
int DiceRoll ();                                /// function to simulate a dice roll
int PlayGame(int hiScore);                      /// Function containing the Game code
int Careerchoice(int diceVal);                  /// function to determine career
based on dice roll
int Salary(int career);                         /// function to determine salary
int Turn(int remCash);                          /// function to simulate each turn
int Invest(int remCash);                        /// function to handle investments
int Growth(int sal);                            /// function to decide Promotion ,
demotion or Stagnancy
int RandomExpense(int remCash);                 /// function to decide Expense
float Taxes(int sal);                           /// function to calculate taxes
int Fortune(int remCash);                       /// function to calculate Fortune
int Invest_in(int remCash , int choice);        /// function to invest in a
particular field
void HighScores();                              /// function to display Score list


void HighScores()
{
    int i = 0 , j ;

    for(i=0 ; i < nPlays ; i++)
    {
        printf("%s : %d",playerList[i],scores[i]);
    }
}


int DiceRoll()    /// Simulates Roll of a dice
{
   time_t t ;
   srand(time(&t));
   int l = ( (rand()%6) + 1 );
   srand(time(&t));
   int h = ( (rand()%6) + 1 );
   return ((l+h) / 2);         /// As dice value starts from 1 and not 0
}

int Careerchoice(int diceVal)   /// Allots Career based on Dice roll
{
    enum career {DOCTOR=1 , ENGINEER , TEACHER , ARTIST , SCIENTIST , MANAGER };
    switch(diceVal)
    {
```

```
64          case 1: return 1;
65          case 2: return 2;
66          case 3: return 3;
67          case 4: return 4;
68          case 5: return 5;
69          case 6: return 6;
70      }
71
72      return 0;
73  }
74  ///
**********************************************************************************************
*************
75
76  int Salary(int career)    /// Allots salary based on Career
77  {
78      switch(career)
79      {
80          case 1: return 50000; break;
81          case 2: return 25000; break;
82          case 3: return 20000; break;
83          case 4: return 25000; break;
84          case 5: return 40000; break;
85          case 6: return 45000; break;
86      }
87      return 0;
88  }
89  ///
**********************************************************************************************
**************
90
91  int Invest_in(int remCash , int choice)    /// Choice based Investment in either
Gold or Real estate or Stocks
92  {
93      time_t x;
94      srand(time(&x));
95      time_t t;
96      srand(time(&t));
97      int l = rand()%5 + 1;
98      srand(time(&t));
99      int h = rand()%5 + 1;
100
101     float  chance = (int)((l+h)/2);           ///  1 in 3 chance of Incurring A loss
102     float luck =  (int)(rand()%5 + 1) ;      /// Return of 0 to 5% profit or loss
103     int investAmt , min;                     /// min is the minimum amt a user must
invest in a particular field
104
105      switch(choice)
106      {
107          case 1:  min = MINGOLD;
108                   if(remCash < MINGOLD)
109                   {
110                       printf("\t\t--You have insufficient funds to invest in Gold
\n");
111                       return remCash;
112                   } break;
113         case 2: min = MINRE;
114                  if(remCash < MINRE)
115                  {
116                      printf("\t\t--You have insufficient funds to invest in Real
estate \n");
117                      Invest(remCash);
118                  } break;
119
120         case 3: min = MINSTOCK;
121                  if(remCash < MINSTOCK)
```

```
122                 {
123                     printf("\t\t--You have insufficient funds to invest in Stock
Market \n");
124                     Invest(remCash);
125                 }
126 
127       }
128 
129       retry:
130       printf("\t\tHow much to invest  ? You will get a return of 0 to 5 percentage
based on your luck \n");
131       scanf("%d",&investAmt );
132 
133       if(investAmt > remCash)    /// Makes sure User doesn't invest what he doesn't
have
134       {
135           printf("\t\tYou cannot invest more than you have \n");
136           goto retry;
137       }
138       else if(investAmt < min)   /// To make sure user invests atleast the minimum
amount
139       {
140           printf("\t\tYou must invest a minimum of %d \n",min);
141           goto retry;
142       }
143 
144 
145       remCash -= investAmt;    /// Subtracts investment Amount from Remaining cash
146 
147       printf("\t\tYou have %d remaining after investing \n",remCash);
148 
149       if(chance == 1)   /// Based on Luck decides whether to incurr a loss or a Profit
150       {
151           investAmt += (investAmt * luck )/ 10;
152           printf("\n\t\t--You have incurred a loss of %d \n\n\n",investAmt);
153           remCash -= investAmt;         /// Subtract the losses from the remaining
cash
154       }
155       else
156       {
157           investAmt += (investAmt * luck) / 10;
158           printf("\n\t\t--Your Investment profit = %d \n\n\n",investAmt);
159           remCash += investAmt;         /// Add the profits to the remaining cash
160 
161       }
162       return remCash;
163 
164  }
165 
166  int Invest(int remCash)    /// Simulates the Investment
167  {
168       int choice ;
169 
170       printf("\t\tYou must invest a minimum of 1000 in Gold \n");
171       printf("\t\tYou must invest a minimum of 10000 in Real Estate \n");
172       printf("\t\tYou must invest a minimum of 3000 in Stock Market \n");
173 
174       choose:
175       printf("\n\t\tPress 1 to invest in Gold\n\t\tPress 2 to Invest in Real
estate\n\t\tPress 3 to invest in Stock Market \n");
176       scanf("%d",&choice);
177 
178       if(choice == 1 || choice == 2 || choice == 3)
179             remCash = Invest_in(remCash , choice);
180       else
181           goto choose;
```

```
182
183
184       return remCash;
185
186   }
187
188   int Turn(int remCash)      /// Provides possible choices that can be made during a
Turn
189   {
190       int choice;
191
192       makechoice:
193       printf(" \n\tPress 1 To Save Money , 2 to Invest \n");
194       scanf("%d",&choice);
195
196       switch(choice)      /// Perform operation based on choice
197       {
198           case 1:  break;
199           case 2:  remCash = Invest(remCash); break;      /// calls the function
invest
200           default : goto makechoice;
201       }
202       return remCash;
203   }
204
205   int Growth(int sal)     /// Decides Growth
206   {
207       time_t t;
208       srand(time(&t));
209
210       switch(rand()%3 + 1)    /// Based on chance decide Promotion , demotion or
Stagnancy
211       {
212           case 1: printf("\t You were promoted and your salary increases by 3000 \n");
213                   sal+=3000;
214                   break;
215           case 2: printf("\t Unfortunately you were demoted and your salary is
decreased by 3000 \n");
216                   sal -= 3000;
217                   break;
218           case 3: printf("\t No Change , Salary remains same \n");
219                   break;
220       }
221
222       return sal;
223   }
224
225   int RandomExpense(int remCash)    /// Generates a random expense
226   {
227       int choice , dice;
228
229       roll:
230       printf("This is a Expense Turn , Roll the dice by pressing 1 \n");
231       scanf("%d",&choice);
232
233       if(choice==1)    /// Makes sure user presses 1 and not any other key
234       {
235           dice = DiceRoll();
236       }
237       else
238       {
239           printf("Enter a valid Input \n");
240           goto roll;
241       }
242
243       switch(dice)  /// Determines expenses based on Roll of a dice
```

```
244      {
245          case 1: printf("You have incurred No expense \n");
246                  break;
247          case 2: printf("You have incurred a Medical expense worth %d \n",(40 *
remCash) / 100);
248                  remCash -= (40 * remCash) / 100;
249                  break;
250          case 3: printf("You have incurred a Fine for A Crime worth %d \n",(20 *
remCash) / 100);
251                  remCash -= (20 * remCash) / 100;
252                  break;
253          case 4: printf("You have incurred Housing Expense worth %d \n",(15 * remCash
) / 100);
254                  remCash -= (15 * remCash) / 100;
255                  break;
256          case 5: printf("You have incurred Banking Expense worth %d \n",(50 * remCash
) / 100 );
257                  remCash -= (50 * remCash) / 100;
258                  break;
259          case 6: printf("You have incurred Vehicular Expense worth %d \n",(30 *
remCash) / 100);
260                  remCash -= (30 * remCash) / 100;
261                  break;
262
263   }
264
265    return remCash;
266  }
267
268  float Taxes(int sal)   /// Calculates Taxes as 15% of Salary
269  {
270      float tax = 0.10 * sal;   /// taxes to be paid are 10% of the salary
271      return tax;
272  }
273
274  int Fortune(int remCash)
275  {
276      int chance , choice , fort;
277
278      roll:
279      printf("There is a 1 in 6 chance of Getting a Fortune \nRoll the dice by
pressing 1 \n");
280      scanf("%d",&choice);
281
282      if(choice == 1)
283          chance = DiceRoll();
284      else
285          goto roll;
286
287      if(chance == 1)
288      {
289          fort = (int) (10.0 * remCash) / 100;                    /// 10% fortune to be
received
290          printf("You have received a fortune of %d \n",fort);
291          remCash += fort;
292          printf("You have %d remaining \n",remCash);
293      }
294      else
295      {
296          printf("Bad Luck ! You have received no fortune \n");
297      }
298
299      return remCash;
300  }
301
302  int PlayGame(int hiScore)                /// hiScore is the highest score achieved
```

```
in the entire history of playing
{
    int  i = 0 ,career , choice, choices , year = 1 , nturn = 1 ;
    int sal , remCash = 0;
    float expense;

    printf("Welcome To Finance of Life \n");
    printf("This Game is designed to give the user a taste of the Outside Financial
World \n");
    printf("Test Your Financial Skills , And see How Good You Are ! \n");

    again:
    printf("Press 1 to play , 2 to View High Scores And 3 to Quit\n");
    scanf("%d",&choices);
    getchar();
    if(choices == 1)
        goto Play;
    else if(choices==2)
        HighScores(nPlays);
    else if(choices == 3)
        goto end;
    else
        goto again;

    Play:
    printf("Enter Your Name \n");

    while( (playerList[nPlays][i] = getchar() )!= '\n')
        i++;

    printf("You will be provided 10 years time to Achieve Victory \n");
    printf("Each year consist of 3 turns \n");
    printf("Roll the dice to choose a career \n");
    printf("Based on Your Die result There are 6 career choices : DOCTOR(1) ,
ENGINEER(2) , TEACHER(3) , POLICE(4) , SCIENTIST(5) , MANAGER(6) \n");

   /* CAREER AND SALARY COMPUTUATION*/
    roll :
    printf("Press 1 to roll the dice \n");
    scanf("%d",&choice);

    if(choice == 1)       /// Makes sure that user presses 1 and not any other key
    {
         career = DiceRoll();
         Careerchoice(career);
    }
    else
    {
        printf("Enter valid number \n");
        goto roll;
    }

    switch(career)                /// Print career
    {
        case 1: printf("\nYou have become an Doctor \n"); break;
        case 2: printf("\nYou have become a Engineer \n"); break;
        case 3: printf("\nYou have become a Teacher \n"); break;
        case 4: printf("\nYou have become an Artist \n"); break;
        case 5: printf("\nYou have become a Scientist \n"); break;
        case 6: printf("\nYou have become a Manager \n"); break;
    }

        sal = Salary(career);      /// Decides salary based on career

    printf("\nYour Salary is %d \n",sal);
    printf("\nYour Base expense based on your salary is 70 percent of your salary
```

```
\n");
366
367      expense = 0.7 * sal;      /// Calculates expenses based on Expenses
368
369      printf("--Your Expense = %.2f \n",expense);
370      printf("--Each year there is a one in three chance of you getting a promotion ,
demotion or stagnancy\n");
371      printf("--REMEMBER YOU WILL LOSE IF YOUR REMAINING CASH REACHES 0 \n");
372      printf("--INVESTMENTS BEAR A 1 IN 3 CHANCE OF A LOSS \n");
373      printf("--TURN 3 IS A RANDOM EXPENSE TURN BE PREPARED FOR IT ! GOOD LUCK !! \n"
);
374      printf("--AT THE END OF TURN 3 YOU WILL BE PROVIDED WITH A CHANCE OF FORTUNE \n"
);
375
376
377      for(year = 1 ; year <= MAXYEAR ; year++)      /// Outer Loop which controls the
Year Going on
378      {
379          expense = 0.7 * sal;
380          remCash += sal;                          /// Before expenses All the salary
is a part of remaining cash
381
382          printf("\n\n------YEAR %d------ \n \n \n",year);
383          printf("\t--This year you have to pay %.2f in taxes \n",Taxes(sal));
384          printf("\t--Your expenses are : %.2f \n",expense);
385          printf("\t--Remaining cash before expenses : %d \n",remCash);
386
387          remCash -=  expense;                     /// To subtract the expenses from
Remaining cash
388
389          printf("\t--Remaining Cash after expenses = %d \n",remCash);
390          printf("\t--Salary at start of year %d = %d \n\n\n",year,sal);
391
392              for(nturn = 1 ; nturn <= MAXTURN ; nturn++)     /// Inner loop to
Control the ongoing turn
393              {
394                  printf("\n\t---Turn %d--- \n",nturn);
395                  remCash = Turn(remCash);                    /// Modify remaining
cash based on Operation performed in The turn
396
397                  if(nturn == 3)                              /// As 3rd turn is a
expense turn
398                  {
399                     remCash =  RandomExpense(remCash);                  /// Allots a
random expense based on dice roll
400
401                      if(remCash <= 0 )            /// Checks if Money = 0 , if so
then player has lost
402                       {
403                            printf("---------------You Lose-------------\n");
404                             goto end;
405                        }
406                  }
407
408                  printf("\t--At the end of Turn %d You Have %d remaining \n \n",nturn
,remCash);
409              }
410
411          remCash = Fortune(remCash);
412          printf("\t--Time to Pay Taxes worth %.2f \n",Taxes(sal));
413          remCash -= Taxes(sal);                                                  ///
To subtract taxes from remaining salary
414          printf("\t--At the end of Year %d You Have %d remaining \n",year,remCash);
415
416           if(remCash <= 0 )           /// Checks if Money = 0 , if so then player
has lost
```

```
417          {
418               printf("-----------------You Lose---------------\n");
419               goto end;
420          }
421
422          sal = Growth(sal);                                                  ///
Modify salary based on promotion , demotion or Stagnancy
423          printf("\t--At the end of year %d , Your salary = %d \n
-------------------------------------------------\n\n\n",year,sal);
424
425
426      }
427
428      scores[nPlays] = remCash;
429      nPlays++;
430
431      end:
432      if(remCash > hiScore)                    /// If high score of previous play is
Lower then modify it
433          hiScore = remCash;
434
435      printf("------------Current Score : %d-------------\n",scores[nPlays-1]);
436      printf("--------------High Score : %d-------------\n",hiScore);
437
438      return hiScore;
439
440  }
441
442
443  #endif // GFILES_H_INCLUDED
```