DATA STRUCTURES PROJECT

FOR B.Tech 3rd sem

SUBMITTED TO: **Ms Sumedha Seniaray**

By :-

Avikal Goel (2K19/MC/028) **Abhijit Barua (2K19/MC/003)**

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Delhi Technological University for their valuable support and providing us a platform to showcase our research and analysis skills in making this project.

We would also like to express our deepest appreciation to all those who provided us with the possibility to complete this project review report. Also, a special gratitude, we want to give our Data Structures Professor Ms Sumedha Seniaray who gave us this opportunity to work on this project and gave us all support and guidance which made us complete the project review duly.

The success and final outcome of this progress report required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of it. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

INDEX

S. No	TOPIC COVERED
1	SOURCE OF MOTIVATION
2	OBJECTIVE
3	BRIEF EXPERIMENTAL DESIGN
4	FURTHER DETAILS ON THE DESIGN
5	RESEARCH METHADOLOGY
6	SCREENSHOTS OF THE CODE
7	SCREENSHOTS OF SOME OUTPUTS
8	FAQ’s
9	RESULTS
10	FURTHER DEVELOPMENTS
11	CONTRIBUTIONS
12	REFERENCES

SOURCE OF MOTIVATION

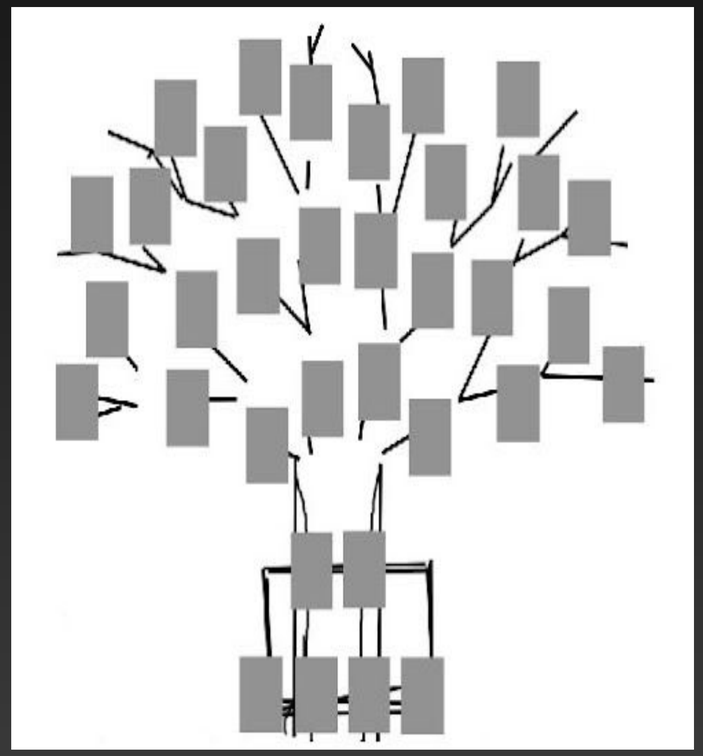
The reason of choosing this as our B.tech MTE Project is simply because of its wide ranging real life applications and importance in different different fields. Moreover, one can always update/modify/change the workings of this project quite easily so that this project can be proved handy for that person too. For example, the same project could be deployed for universities and on the other hand for investigation purposes.

Also, one of the good reason for selecting this project out of many others for us was to learn and implement efficient memory usage too in every aspects and sharpen our concepts, when it comes to optimisations in time complexities. By this project, we also aspired to learn techniques for storing such wide data into as efficient way as possible.

Hence, concluding, this project is overall a good learning experience for us and also a good one, when it comes to real world applications.

OBJECTIVE

GOAL: The project focuses on the human nature that many a times, we remember several traits of a person, including his approximate height, his approximate age, his profession, his achievements, his gender, the era to which the person belongs, his nationality etc, but tend to forget other very important details. This project is the solution to such a problem.



Just like the tree which is being shown in the diagram, the code decides the flow of controller and tries to reach the leaf nodes according to the questions asked and the answers received.

Once leaf node is reached, it becomes the answer of the code and is the most suitable personality out of the complete database the user wanted to know about.

BRIEF EXPERIMENTAL DESIGN

Our code implements a binary tree with some extra characteristics.

A **binary tree** is a hierarchical data structure in which each node has at most two children generally referred as left child and right child. Each node contains three components: Pointer to left subtree. Pointer to right subtree. Data element.

This project takes several inputs from the user in the form of true/false questions. These inputs decide the flow of code and hence finally, from the database, the code searches for the person who matches with the approximate details given by the user.

FURTHER DETAILS ON THE DESIGN

NOTE:- THIS IMPLEMENTATION IS ONLY FOR DEMONSTRATION POINT OF VIEW. ONE CAN ALWAYS CHANGE THE OUTPUTS OF THE CODE ACCORDING TO HIS NEEDS, KEEPING THE OVERALL SKELATON OF THE CODE SAME AS THIS ONE.

First of all, initial database is being prepared, which consists of several famous personalities from across the world, whether alive or dead. Their data is being collected and is stored efficiently in tree data structure. Now the code is ready with ample amount of pre-recorded data of several famous personalities. On startup, user is being asked whether they want to execute the code and find the desired personality, if user answers a YES, the code is further processed and user is now asked different basic-basic **true/false type questions**. Once all the question are answered (for some questions to which user isn't aware of the data, he might skip those questions), then the code performs several binary searches and finally get the best possible personality out of the full database which might match to the user’s person which he thought about.

RESEARCH-METHADOLOGY

The single most challenging work to implement the project was to prepare the dataset nicely and wisely. This was the most crucial part of the whole project. This is because we had to choose limited number of characters out of millions who have recognition in any field. Also we had to choose characters so that we cover vast variety of users. Like it shouldn't be the case that out of 500 characters added to the database, 400 are sportsmen. Hence to avoid such cases, we did the following measure:-

We consulted many different people, all of different areas and different fields or professions and asked them to run the program and find out whether their character was present in the database or not. At this time, we had quite a few characters in the database. As we performed this activity on more and more people, we got more and more new characters for our database, which were later added in the mainstream of the code. Hence, after a span of time, we were ready with a much better database, which was fulfilling a much wider set of people and answered correctly to their query.

SCREENSHOTS OF THE CODE

```cpp
int main()
 {
     cout<<"                WELCOME TO THE PROGRAM:  YOU ARE FOUND   \n\n\n\n" ;

     cout<<"here, the program will ask you simple questions regarding the basic properties of the
         person and hence will try to figure out the character going in in your mind\n\n\n" ;
     cout<<"HOPE WE GUESS THE CHARACTER CORRECTLY\n\n\n\n\n\n\n" ;


     cout<<"So here we begin\n" ;
     cout<<"Press 1 to continue\n" ;
     int temp ;
     cin>>temp ;
     if(temp!=1) { cout<<"OHH NOOO, WE CANT CONTINUE \n" ; return 0 ; }

     cout<<"\n\n" ;
     cout<<"Basic Rules:\n" ;
     cout<<". You will be asked simple YES/NO questions\n" ;
     cout<<". If you dont know the answer, reply with 0\n" ;
     cout<<". If your answer is NO, reply with 1\n" ;
     cout<<". Finally, if your answer is YES, reply with 2\n" ;
     cout<<"\n. After a sequence of questions, we shall reach to the character you wished to search
         for and hence provide other info of that character too\n\n" ;
     |

     start() ;



    return 0;
}
```

INT MAIN FUNCTION

This tells the overall rules to follow so that program gives correct output

```cpp
void start()
{
    vector<string>ans ;
    f1(ans) ;

    cout<<"\n\n\n" ;
    delay(2000) ;
    cout<<"(thinking)\n.\n.\n" ;
    delay(2000) ;
    cout<<"(still thinking)\n.\n.\n" ;
    delay(2000) ;
    cout<<"(compiling our results)\n\n\n" ;
    delay(2000) ;
    cout<<"(compilation done successfully)\n" ;
    cout<<"OHH YESS, WE GOT OUR ANSWER\n\n" ;
    delay(4000) ;
    if( ans.empty() )
    {
        cout<<"OOPS!! YOU THOUGHT OF A RARE CHARACTER AND HENCE BEATED THE MACHINE FROM YOUR GENERAL
            KNOWLEDGE\n\n" ;
    }
    cout<<"BASED ON YOUR GIVEN ANSWERS, THESE ARE THE CHARACTERS YOU PROBABLY WOULD HAVE THOUGHT
        OF:\n\n" ;

    for( int i=0; i<ans.size(); i++ )
    {
        cout<<i+1<<")  " ;
        cout<<ans[i]<<"\n" ;
    }
    cout<<"\n\n" ;
    cout<<"ENTER 1 IF YOUR CHARCTER IF FOUND, ELSE ENTER 0\n" ;
    int r ; cin>>r ;
    cout<<"\n\n" ;
    if(r==0)
    {
        cout<<"OOPS!! YOU THOUGHT OF A RARE CHARACTER AND HENCE BEATED THE MACHINE FROM YOUR GENERAL
            KNOWLEDGE\n\n" ;
    }
    if(r==1)
    {
        cout<<"NICE TO KNOW THAT WE GOT TO YOUR CHARACTER CORRECTLY\n\n" ;
    }
```

START FUNCTION

This is the start functions. This marks the beginning of recursion

```
1360  void delay(int number_of_seconds)
1361  {
1362      // Converting time into milli_seconds
1363      int milli_seconds = 1000 * number_of_seconds;
1364
1365      // Storing start time
1366      clock_t start_time = clock();
1367
1368      // looping till required time is not achieved
1369      while (clock() < start_time + milli_seconds)
1370          ;
1371  }
```

VOID DELAY FUNCTION

This function delays the running of the code to make it a little more attractive

```
void f4(vector<string>&ans)
{
    int r;
    cout<<"Is your character INDIAN??\n" ;
    cin>>r ;
    if(r==1) { f8(ans) ;return ;  }
    if(r==2) { f9(ans); return ; }
    f8(ans) ; f9(ans) ;
}
void f3( vector<string>&ans )
{
    cout<<"Is your character alive??\n";
    int r ; cin>>r ;
    if(r==1){ f6(ans) ; return ; }
    if(r==2) { f7(ans) ; return ;  }
    f6(ans); f7(ans) ;
}
void f2( vector<string>&ans )
{
    cout<<"Is your character alive??\n";
    int r ; cin>>r ;
    if(r==1){ f4(ans) ; return ; }
    if(r==2) { f5(ans) ; return ;  }
    f4(ans); f5(ans) ;
}
void f1( vector<string>&ans )
{
    cout<<"So here we begin by the first question\n" ;
    int r ;
    cout<<"Is your character a female??\n" ;
    cin>>r ;

    if( r==1 ) { f2(ans) ; return ; }
    if( r==2 ) { f3(ans) ; return ;  }
    f2(ans) ; f3(ans) ;
}
```

```
void f7(vector<string>&ans)
{
    int r;
    cout<<"Is your character INDIAN??\n" ;
    cin>>r ;
    if(r==1) { f14(ans) ;return ;  }
    if(r==2) { f15(ans); return ; }
    f14(ans) ; f15(ans) ;
}
void f6(vector<string>&ans)
{
    int r;
    cout<<"Is your character INDIAN??\n" ;
    cin>>r ;
    if(r==1)
    {
        string s="MARGARET THATCHER" ; ans.push_back(s)
        s="MARIE CURIE" ; ans.push_back(s) ;
        return ;
    }
    if(r==2) { f13(ans); return ; }
    string s="MARGARET THATCHER" ; ans.push_back(s) ;
    s="MARIE CURIE" ; ans.push_back(s) ;
    f13(ans) ;
}

void f5(vector<string>&ans)
{
    int r;
    cout<<"Is your character INDIAN??\n" ;
    cin>>r ;
    if(r==1) { f10(ans) ;return ;  }
    if(r==2) { f11(ans); return ; }
    f10(ans) ; f11(ans) ;
}
```

RECURSIVE FUNCTIONS

```
1248 void f11( vector<string>&ans )
1249 {
1250     int r ;
1251     cout<<" Is your character an actor??\n " ;
1252     cin>>r ;
1253     if( r==1 ) { f22(ans) ; return ; }
1254     if( r==2 ) { f23(ans) ; return ; }
1255     f22(ans) ; f23(ans) ;
1256 }
1257 void f10( vector<string>&ans )
1258 {
1259     cout<<"Has your character ever been president/prime-minister of his nation??\n" ;
1260     int r; cin>>r ;
1261     if( r==1 ) { f20(ans) ; return ;  }
1262     if(r==2) { f21(ans)  ; return ;  }
1263     f20(ans) ; f21(ans) ;
1264 }
1265 void f9( vector<string>&ans )
1266 {
1267     int r ;
1268     cout<<" Is your character associated with British-Indian struggle??\n " ;
1269     cin>>r ;
1270     if( r==1 ) { f18(ans) ; return ; }
1271     if( r==2 ) { f19(ans) ; return ; }
1272     f18(ans) ; f19(ans) ;
1273 }
1274 void f8( vector<string>&ans )
1275 {
1276     cout<<"Is your character a scientist/mathematician/inventor or from any such research oriented
        profession??\n" ;
1277     int r ; cin>>r ;
1278     if(r==1)
1279     {
1280         f16(ans) ; return ;
1281     }
1282     if(r==2)
1283     {
1284         f17(ans) ; return ;
1285     }
1286     f16(ans) ; f17(ans) ;
1287 }
```

```
1208 void f15(vector<string>&ans)
1209 {
1210     cout<<"Is the character a politician??\n" ;
1211     int r ; cin>>r ;
1212     if(r==1) { f30(ans) ; return ;  }
1213     if(r==2)
1214     {
1215         string s="SONIA GANDHI" ; ans.push_back(s) ;
1216         s="MAMTA BANERJEE" ; ans.push_back(s) ;
1217         s="NIRMALA SITARAMAN" ;  ans.push_back(s) ;
1218         return ;
1219     }
1220     f30(ans) ;
1221     string s="SONIA GANDHI" ; ans.push_back(s) ;
1222     s="MAMTA BANERJEE" ; ans.push_back(s) ;
1223     s="NIRMALA SITARAMAN" ;  ans.push_back(s) ;
1224 }
1225 void f14( vector<string>&ans )
1226 {
1227     cout<<"Is the character an actress??\n" ;
1228     int r ; cin>>r ;
1229     if(r==1)
1230     {
1231         f28(ans) ; return ;
1232     }
1233     if( r==2 )
1234     {
1235         f29(ans) ; return ;
1236     }
1237     f28(ans) ; f29(ans) ;
1238 }
1239 void f13( vector<string>&ans )
1240 {
1241     int r ;
1242     cout<<" Is your character associated with British-Indian struggle??\n " ;
1243     cin>>r ;
1244     if( r==1 ) { f26(ans) ; return ; }
1245     if( r==2 ) { f27(ans) ; return ; }
1246     f26(ans) ; f27(ans) ;
1247 }
```

RECURSIVE FUNCTIONS

```
1166 void f17( vector<string>&ans )
1167 {
1168     cout<<"Has your character contributed something in the fields of physics??\n" ;
1169     int r ; cin>>r;
1170     if(r==1)
1171     {
1172         string s="BLAISE PASCAL" ; ans.push_back(s) ;
1173         s="CHARLES DARWIN"  ; ans.push_back(s) ;
1174         return;
1175     }
1176     if(r==2)
1177     {
1178         f35(ans) ; return;
1179     }
1180     string s="BLAISE PASCAL" ; ans.push_back(s) ;
1181     s="CHARLES DARWIN"  ; ans.push_back(s) ;
1182     f35(ans) ;
1183 }
1184 void f16( vector<string>&ans )
1185 {
1186     cout<<"Is your character from mughal dynasty??\n" ;
1187     int r ; cin>>r ;
1188     if(r==1)
1189     {
1190         f32(ans) ; return ;
1191     }
1192     if(r==2)
1193     {
1194         string s="AKBAR" ; ans.push_back(s) ;
1195         s="BAIRAM KHAN" ; ans.push_back(s) ;
1196         s="BABUR" ; ans.push_back(s) ;
1197         s="BAHADUR SHAH ZAFAR" ; ans.push_back(s) ;
1198         s="SHAH JAHAN" ; ans.push_back(s) ;
1199         return ;
1200     }
1201     f32(ans) ;
1202     string s="AKBAR" ; ans.push_back(s) ;
1203     s="BAIRAM KHAN" ; ans.push_back(s) ;
1204     s="BAHADUR SHAH ZAFAR" ; ans.push_back(s) ;
1205     s="BABUR" ; ans.push_back(s) ;
1206     s="SHAH JAHAN" ; ans.push_back(s) ;
1207 }
```

```
1141 void f20( vector<string>&ans )
1142 {
1143     cout<<"Does the character has international recognition in any sports/game??\n" ;
1144     int r ; cin>>r ;
1145     if(r==1) { f40(ans) ; return ; }
1146     if(r==2) { f41(ans) ; return ;  }
1147     f40(ans) ;  f41(ans) ;
1148 }
1149 void f19(vector<string>&ans)
1150 {
1151     int r ;
1152     cout<<"Was the character hanged due to bombarding in the assembly??\n" ;
1153     cin>>r ;
1154     if(r==1) { f38(ans) ; return ;  }
1155     if( r==2 ) { f39(ans) ; return ;  }
1156     f38(ans) ;   f39(ans) ;
1157 }
1158 void f18( vector<string>&ans )
1159 {
1160     cout<<"Is your character a warrior or closely related to a famous warrior or a king??\n" ;
1161     int r ; cin>>r ;
1162     if(r==1) { f36(ans) ; return ;  }
1163     if( r==2 ) { f37(ans) ; return ; }
1164     f36(ans) ; f37(ans) ;
1165 }
```

RECURSIVE FUNCTIONS

```
void f22( vector<string>&ans )
{
    int r ;
    cout<<" Is your character a sportsman??\n " ;
    cin>>r ;
    if( r==1 ) { f44(ans) ; return ; }
    if( r==2 ) { f45(ans) ; return ; }
    f44(ans) ; f45(ans) ;
}
void f21( vector<string>&ans )
{
    cout<<"Is your character currently also a prime minister/president\n"  ;
    int r; cin>>r ;
    if(r==1)
    {
        string s="BARACK OBAMA" ; ans.push_back(s) ;
        s="BILL CLINTON" ; ans.push_back(s) ;
        s= "SHINZO ABE" ;  ans.push_back(s) ; return ;
    }
    if(r==2)
    {
        string s="DONALD TRUMP" ; ans.push_back(s) ;
        s="XI JINPING" ; ans.push_back(s) ;
        s="VLADIMIR PUTIN" ; ans.push_back(s) ;
        s="IMRAN KHAN" ; ans.push_back(s) ;
        return ;
    }
    string s="BARACK OBAMA" ; ans.push_back(s) ;
    s="BILL CLINTON" ; ans.push_back(s) ;
    s="IMRAN KHAN" ; ans.push_back(s) ;
    s= "SHINZO ABE" ;  ans.push_back(s) ;
    s="VLADIMIR PUTIN" ; ans.push_back(s) ;
    s="DONALD TRUMP" ; ans.push_back(s) ;
    s="XI JINPING" ; ans.push_back(s) ;
}
```

```
void f27( vector<string>&ans )
{
    int r ;
    cout<<"Has the character been the prime minister of India??\n" ;
    string s="INDIRA GANDHI"  ;
    cin>>r ;
    if(r==1) { f54(ans) ; return ;  }
    if( r==2 ) { ans.push_back(s) ;  return;  }
    ans.push_back(s) ;   f54(ans) ;

}
void f26( vector<string>&ans )
{
    cout<<"Is the character an astranaut??\n" ;
    int r ; cin>>r ;
    if( r==2 )
    {
        string s="KALPANA CHAWLA" ; ans.push_back(s) ; return ;
    }
    if(r==1)
    {
        string s="QUEEN JODHA" ; ans.push_back(s) ;
        s="SUSHMA SWARAJ" ; ans.push_back(s) ;
        return ;
    }
    string s="KALPANA CHAWLA" ; ans.push_back(s) ;
    s="QUEEN JODHA" ; ans.push_back(s) ;
    s="SUSHMA SWARAJ" ; ans.push_back(s) ;
}
void f23( vector<string>&ans )
{
    cout<<"Is your character's father also closely relatd to film industry??\n" ;
    int r; cin>>r ;
    if( r==1 ) { f46(ans) ; return ;  }
    if(r==2) { f47(ans) ; return ;  }
    f46(ans) ; f47(ans) ;
}
```

RECURSIVE FUNCTIONS

```
void f30( vector<string>&ans )
{
    cout<<"Is the character an actress??\n" ;
    int r ; cin>>r ;
    if(r==1) { f60(ans) ; return ;  }
    if( r==2 ) { f61(ans) ; return ;  }
    f60(ans) ;   f61(ans) ;
}
void f29( vector<string>&ans )
{
    cout<<"Is the character born in Sri Lanka??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        string s= "ANGELINA JOLIE" ; ans.push_back(s) ;
         s="EMMA STONE" ; ans.push_back(s) ;
         s="JENNIFER LAWRENCE" ; ans.push_back(s) ; return ;
    }
    if( r==2 ) { string s="JACQUELINE FERNANDEZ" ; ans.push_back(s) ; return;  }
    string s= "ANGELINA JOLIE" ; ans.push_back(s) ;
     s="EMMA STONE" ; ans.push_back(s) ;
     s="JENNIFER LAWRENCE" ; ans.push_back(s) ;
    s="JACQUELINE FERNANDEZ" ; ans.push_back(s) ;
}
void f28( vector<string>&ans )
{
    cout<<"Is your character from tech profile??\n" ;
    int r ; cin>>r ;
    if( r==1 ) { f56(ans) ; return ; }
    if(r==2)
    { string s="MEG WHITMAN" ; ans.push_back(s) ;
        s="MELINDA GATES" ; ans.push_back(s) ;
        return;
    }
    f56(ans) ; string s="MEG WHITMAN" ; ans.push_back(s) ;
    s="MELINDA GATES" ; ans.push_back(s) ;
}
```

```
void f35( vector<string>&ans )
{
    cout<<"Is the character famous for suggesting the model of an atom
        (protons,neutrons,eletrons)??\n" ;
    int r ; cin>>r ;
    if(r==1) { f70(ans) ; return ;  }
    if( r==2 )
    {
        string s="NEIL BOHR" ; ans.push_back(s);
        s="RUTHETFORD" ; ans.push_back(s) ;
        return ;
    }
    f70(ans) ;
    string s="NEIL BOHR" ; ans.push_back(s);
    s="RUTHETFORD" ; ans.push_back(s) ;
}
void f32( vector<string>&ans )
{
    cout<<"Did your character die due to drugs??\n" ;
    int r ; cin>>r ;
    if(r==2)
    {
        string s="MICHAEL JACKSON" ; ans.push_back(s) ; return ;
    }
    if(r==1)
    {
        f64(ans) ; return ;
    }
    string s="MICHAEL JACKSON" ; ans.push_back(s) ;   f64(ans) ;
}
```

RECURSIVE FUNCTIONS

```
void f38( vector<string>&ans )
{
    cout<<"Was the person the founder and leader of INDIAN NATIONAL ARMY??\n";
    int r ; cin>>r ;
    string s="SUBHASH CHANDRA BOSE" ;
    if(r==1){ f76(ans) ; return ; }
    if( r==2 ) { ans.push_back(s) ; return ; }
    ans.push_back(s) ;   f76(ans) ;
}
void f37( vector<string>&ans )
{
    cout<<"Is your character a RAJPUT warrior??\n" ;
    int r; cin>>r ;
    if(r==1){ f74(ans) ; return ; }
    if(r==2)
    {
        string s="MAHARANA PRATAP" ; ans.push_back(s) ;
        s="MAHARANA UDAY SINGH" ;ans.push_back(s) ; return ;
    }
    f74(ans) ;
    string s="MAHARANA PRATAP" ; ans.push_back(s) ;
    s="MAHARANA UDAY SINGH" ;ans.push_back(s) ;
}
void f36( vector<string>&ans )
{
    cout<<"Has the character served INDIA as prime-minister or a president ??\n"
    int r ; cin>>r ;
    if(r==1) { f72(ans) ; return ;  }
    if( r==2 )
    {
        string s="Dr. APJ ABDUL KALAAM" ; ans.push_back(s) ;
        s="ATAL BIHARI VAJPAYEE" ; ans.push_back(s) ;
        s="LAL BAHADUR SHASTRI" ; ans.push_back(s) ;
        return ;
    }
    f72(ans) ;
    string s="Dr. APJ ABDUL KALAAM" ; ans.push_back(s) ;
    s="ATAL BIHARI VAJPAYEE" ; ans.push_back(s) ;
    s="LAL BAHADUR SHASTRI" ; ans.push_back(s) ;
}
```

```
void f40( vector<string>&ans )
{
    cout<<"Does the character work at a good reputed tech company or has a significant contribution in
        tech field??\n" ;
    int r ; cin>>r ;
    if(r==1) { f80(ans) ; return;  }
    if(r==2) { f81(ans) ; return;  }
    f80(ans) ;  f81(ans) ;
}
void f39( vector<string>&ans )
{
    cout<<"Was the character the main leader of the 3 hanged??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        string s="SHIVARAM RAJGURU" ;
        ans.push_back(s);
        s="SUKHDEV THAPAR" ;
        ans.push_back(s) ;
        return ;
    }
    if( r==2 )
    {
        string s="BHAGAT SINGH" ;
        ans.push_back(s);
        return ;
    }
    string s="SHIVARAM RAJGURU" ;
    ans.push_back(s);
    s="SUKHDEV THAPAR" ;
    ans.push_back(s) ;
    s="BHAGAT SINGH" ;
    ans.push_back(s);
}
```

RECURSIVE FUNCTIONS

```
903 void f44( vector<string>&ans )
904 {
905     cout<<"Is the character a politician??\n" ;
906     int r ; cin>>r ;
907     if(r==1) { f88(ans) ; return ; }
908     if(r==2) { f89(ans) ; return ;  }
909     f88(ans) ;  f89(ans) ;
910 }
911 void f41( vector<string>&ans )
912 {
913     cout<<"Is your character a cricketer??\n" ;
914     int r ; cin>>r ;
915     if(r==1) { f82(ans) ; return ; }
916     if( r==2 )
917     {
918         string s="KIERON POLLARD" ; ans.push_back(s) ;
919         s="CHRIS GAYLE" ; ans.push_back(s) ;
920         s="DAVID WARNER" ; ans.push_back(s) ;
921         s="A B D VILLIERS" ; ans.push_back(s) ;
922         return ;
923     }
924     f82(ans) ;
925     string s="KIERON POLLARD" ; ans.push_back(s) ;
926     s="CHRIS GAYLE" ; ans.push_back(s) ;
927     s="DAVID WARNER" ; ans.push_back(s) ;
928     s="A B D VILLIERS" ; ans.push_back(s) ;
929 }
```

```
881 void f45( vector<string>&ans )
882 {
883     cout<<"Is the character a cricketer??\n" ;
884     int r; cin>>r ;
885     if(r==1)
886     {
887         string s="MILKHA SINGH" ; ans.push_back(s) ;
888         s="MAJOR DHYAN CHAND"  ; ans.push_back(s) ;
889         s="VISHWANATHAN ANAND"  ; ans.push_back(s) ;
890         s="THE GREAT KHALI"  ; ans.push_back(s) ;
891         return ;
892     }
893     if( r==2 )
894     {
895         f91(ans) ; return ;
896     }
897     string s="MILKHA SINGH" ; ans.push_back(s) ;
898     s="VISHWANATHAN ANAND"  ; ans.push_back(s) ;
899     s="THE GREAT KHALI"  ; ans.push_back(s) ;
900     s="MAJOR DHYAN CHAND"  ; ans.push_back(s) ;
901     f91(ans) ; return ;
902 }
```

```
858 void f46( vector<string>&ans )
859 {
860     cout<<"Is the character a youngster in the industry??\n" ;
861     int r ; cin>>r ;
862     if(r==1)
863     {
864         string s="SHAHRUKH KHAN" ; ans.push_back(s) ;
865         s="AMITABH BACHCHAN" ; ans.push_back(s) ;
866         return;
867     }
868     if(r==2)
869     {
870         string s="AYUSHMAAN KHURRANA" ; ans.push_back(s) ;
871         s="KARTIK ARYAN" ; ans.push_back(s) ;
872         s="VICKY KAUSHAL"; ans.push_back(s) ;
873         return;
874     }
875     string s="SHAHRUKH KHAN" ; ans.push_back(s) ;
876     s="AMITABH BACHCHAN" ; ans.push_back(s) ;
877     s="AYUSHMAAN KHURRANA" ; ans.push_back(s) ;
878     s="KARTIK ARYAN" ; ans.push_back(s) ;
879     s="VICKY KAUSHAL"; ans.push_back(s) ;
880 }
```

RECURSIVE FUNCTIONS

```
void f56( vector<string>&ans )
{
    cout<<"Is your character a saint??\n" ;
    int r ; cin>>r ;
    if(r==1) { f112(ans) ; return ; }
    if(r==2) { string s="MOTHER TERESA" ; ans.push_back(s) ; return ;  }
    string s="MOTHER TERESA" ; ans.push_back(s) ;
    f112(ans) ;
}
void f54( vector<string>&ans )
{
    int r  ;
    cout<<"Is the character a well known queen of a territory??\n" ;
    string s1="SAROJINI NAIDU" ;
    string s2="RANI LAKSHMIBAI" ;
    cin>>r ;
    if(r==1) {  ans.push_back(s1) ; return ;  }
    if( r==2 ) { ans.push_back(s2) ; return ; }
    ans.push_back(s1) ; ans.push_back(s2) ;
}
void f47( vector<string>&ans )
{
    cout<<"Is th character famous for performing stunts??\n" ;
    int r ; cin>>r ;
    if( r==1 )
    {
        string s="SALMAN KHAN" ; ans.push_back(s) ;
        s="VARUN DHAWAN" ; ans.push_back(s) ;
        s="AAMIR KHAN" ; ans.push_back(s) ; return ;
    }
    if(r==2)
    {
        string s="HRITIK ROSHAN" ; ans.push_back(s) ;
        s="TIGER SHROFF" ; ans.push_back(s) ;
        return ;
    }
    string s="SALMAN KHAN" ; ans.push_back(s) ;
    s="VARUN DHAWAN" ; ans.push_back(s) ;
    s="TIGER SHROFF" ; ans.push_back(s) ;
    s="AAMIR KHAN" ; ans.push_back(s) ;
    s="HRITIK ROSHAN" ; ans.push_back(s) ;
}
```

```
void f64( vector<string>&ans )
{
    cout<<"Has your character faced paralysis??\n";
    int r; cin>>r ;
    if(r==2)
    {
        string s="STEPHEN HAWKING" ; ans.push_back(s) ; return;
    }
    if(r==1)
    {
        f128(ans) ; return;
    }
    string s="STEPHEN HAWKING" ; ans.push_back(s) ;  f128(ans) ;
}
void f61( vector<string>&ans )
{
    cout<<"Is the character married??\n" ;
    int r ; cin>>r ;
    if(r==1) { f122(ans) ; return ; }
    if(r==2) { f123(ans) ; return ;  }
    f122(ans) ; f123(ans) ;
}
void f60( vector<string>&ans )
{
    cout<<"Is the character a singer??\n" ;
    int r ; cin>>r ;
    if(r==1) { f120(ans) ; return ; }
    if(r==2)
    {
        string s="NEHA KAKKAR" ; ans.push_back(s) ;
        s="SUNIDHI CHAUHAN" ; ans.push_back(s) ;
        s="SHREYA GHOSHAL" ; ans.push_back(s) ;
        s="LATA MANGESHKAR" ; ans.push_back(s) ;
        return ;
    }
    f120(ans) ;
    string s="NEHA KAKKAR" ; ans.push_back(s) ;
    s="SUNIDHI CHAUHAN" ; ans.push_back(s) ;
    s="SHREYA GHOSHAL" ; ans.push_back(s) ;
    s="LATA MANGESHKAR" ; ans.push_back(s) ;
}
```

```
void f72( vector<string>&ans )
{
    cout<<"Did your character experianced an unnatural death??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        string s="SIR CV RAMAN" ; ans.push_back(s);
        s="HARIVANSH RAI BACHCHAN" ; ans.push_back(s);
        return;
    }
    if(r==2)
    {
        string s="SUSHANT SINGH RAJPUT" ; ans.push_back(s); return;
    }
    string s="SIR CV RAMAN" ; ans.push_back(s) ;
    s="HARIVANSH RAI BACHCHAN" ; ans.push_back(s);
    s="SUSHANT SINGH RAJPUT" ; ans.push_back(s);
}
void f70( vector<string>&ans )
{
    cout<<"Is the character famous for the 3 laws of motion??\n" ;
    int r;
    cin>>r ;
    if(r==1) { f140(ans) ; return ; }
    if(r==2)
    {
        string s="SIR ISAAC NEWTON" ; ans.push_back(s) ; return ;
    }
    f140(ans) ;
    string s="SIR ISAAC NEWTON" ; ans.push_back(s) ;
}
```

RECURSIVE FUNCTIONS

```
717 void f76( vector<string>&ans )
718 {
719     cout<<"Did the character later on became first president or prime minister??\n";
720     int r ; cin>>r;
721     if(r==1) { f152(ans) ; return ;  }
722     if( r==2 ) { f153(ans) ; return;  }
723     f152(ans) ;  f153(ans) ;
724 }
725 void f74( vector<string>&ans )
726 {
727     cout<<"Is your character associated to MAURYAN dynasty??\n" ;
728     int r;cin>>r ;
729     if(r==1)
730     {
731         string s="SHIVAJI" ; ans.push_back(s) ; return ;
732     }
733     if(r==2)
734     {
735         string s="CHANDRAGUPTA MAURYA" ; ans.push_back(s)  ;
736         s="ASHOKA" ; ans.push_back(s)  ;
737         s="CHANAKYA" ; ans.push_back(s)  ;  return ;
738     }
739     string s="SHIVAJI" ; ans.push_back(s) ;
740     s="ASHOKA" ; ans.push_back(s)  ;
741     s="CHANDRAGUPTA MAURYA" ; ans.push_back(s)  ;
742     s="CHANAKYA" ; ans.push_back(s)  ;
743 }
```

```
696 void f80( vector<string>&ans )
697 {
698     cout<<"Is your character a singer??\n" ;
699     int r ;cin>>r ;
700     if(r==2)
701     {
702         string s="JUSTIN BIEBER" ; ans.push_back(s) ;
703         s="TOM WALKER" ; ans.push_back(s) ;
704         s="SHAWN MENDES" ; ans.push_back(s) ;
705         return ;
706     }
707     if(r==1)
708     {
709         string s="WARREN BUFFET" ; ans.push_back(s) ;
710         return ;
711     }
712     string s="JUSTIN BIEBER" ; ans.push_back(s) ;
713     s="WARREN BUFFET" ; ans.push_back(s) ;
714     s="TOM WALKER" ; ans.push_back(s) ;
715     s="SHAWN MENDES" ; ans.push_back(s) ;
716 }
```

RECURSIVE FUNCTIONS

```
671 void f81( vector<string>&ans )
672 {
673     cout<<"Is your character presently or at any point of time a CEO or any other position of honour
        in any social media application??\n" ;
674     int r ; cin>>r ;
675     if(r==1)
676     {
677         string s="JEFF BEZOS" ; ans.push_back(s) ;
678          s="TIM COOK" ; ans.push_back(s) ;
679         s="ELON MUSK" ; ans.push_back(s) ;
680         s="BILL GATES" ; ans.push_back(s) ;
681         return;
682     }
683     if( r==2 )
684     {
685         string s="MARK ZUCKERBURG" ; ans.push_back(s) ;
686         s="JEFF WEINER" ; ans.push_back(s) ;
687         return ;
688     }
689     string s="JEFF BEZOS" ; ans.push_back(s) ;
690      s="TIM COOK" ; ans.push_back(s) ;
691     s="ELON MUSK" ; ans.push_back(s) ;
692     s="MARK ZUCKERBURG" ; ans.push_back(s) ;
693     s="JEFF WEINER" ; ans.push_back(s) ;
694     s="BILL GATES" ; ans.push_back(s) ;
695 }
```

```
644 void f82( vector<string>&ans )
645 {
646     cout<<"Is the character a footballer??\n" ;
647     int r ; cin>>r ;
648     if(r==1)
649     {
650         string s="MAGNUS CARLSEN" ; ans.push_back(s) ;
651         s="USAIN BOLT" ; ans.push_back(s) ;
652         s="ROGER FEDERER" ; ans.push_back(s) ;
653         s="RAFAEL NADAL" ; ans.push_back(s) ;
654         return ;
655     }
656     if(r==2)
657     {
658         string s="CRISTIANO RONALDO" ; ans.push_back(s) ;
659         s="LIONEL MESSI" ; ans.push_back(s) ;
660         s="NEYMAR JUNIOR" ; ans.push_back(s) ;
661         return ;
662     }
663     string s="MAGNUS CARLSEN" ; ans.push_back(s) ;
664     s="USAIN BOLT" ; ans.push_back(s) ;
665     s="ROGER FEDERER" ; ans.push_back(s) ;
666     s="RAFAEL NADAL" ; ans.push_back(s) ;
667     s="CRISTIANO RONALDO" ; ans.push_back(s) ;
668     s="LIONEL MESSI" ; ans.push_back(s) ;
669     s="NEYMAR JUNIOR" ; ans.push_back(s) ;
670 }
```

RECURSIVE FUNCTIONS

```
620 void f91( vector<string>&ans )
621 {
622     cout<<"Does the character still play international cricket??\n" ;
623     int r; cin>>r ;
624     if(r==1) { f182(ans) ; return ; }
625     if(r==2) { f183(ans) ; return ;  }
626     f182(ans) ; f183(ans) ;
627 }
628 void f89( vector<string>&ans )
629 {
630     cout<<"Is your character a part of current cabinet team of ministers??\n";
631     int r ; cin>>r;
632     if(r==1) { f178(ans) ; return ;  }
633     if( r==2 ) { f179(ans) ; return;  }
634     f178(ans) ;  f179(ans) ;
635 }
636 void f88( vector<string>&ans )
637 {
638     cout<<"Is the character at a reputed post in a big company??\n" ;
639     int r ; cin>>r ;
640     if(r==1) { f176(ans) ; return ; }
641     if(r==2) { f177(ans) ; return ;  }
642     f176(ans) ;  f177(ans) ;
643 }
```

```
583 void f120( vector<string>&ans )
584 {
585     cout<<"Is your character a sportsperson??\n";
586     int r ;cin>>r ;
587     if(r==1)
588     {
589         string s="NITA AMBANI" ; ans.push_back(s) ;
590         return;
591     }
592     if(r==2) { f241(ans) ; return ; }
593     f241(ans) ;
594     string s="NITA AMBANI" ; ans.push_back(s) ;
595 }
596 void f112( vector<string>&ans )
597 {
598     cout<<"Is your character a singer??\n" ;
599     int r; cin>>r ;
600     if(r==1)
601     { string s ;
602         s="HILLARY CLINTON" ; ans.push_back(s) ;
603         return ;
604     }
605     if(r==2)
606     {
607         string s="MADONNA" ; ans.push_back(s) ;
608         s="TAYLOR SWIFT" ; ans.push_back(s) ;
609         s="SELENA GOMEZ" ; ans.push_back(s) ;
610         s="JENNIFER LOPEZ" ; ans.push_back(s) ;
611         return ;
612     }
613     string s ;
614     s="HILLARY CLINTON" ; ans.push_back(s) ;
615     s="MADONNA" ; ans.push_back(s) ;
616     s="SELENA GOMEZ" ; ans.push_back(s) ;
617     s="JENNIFER LOPEZ" ; ans.push_back(s) ;
618     s="TAYLOR SWIFT" ; ans.push_back(s) ;
619 }
```

```
564 void f123( vector<string>&ans )
565 {
566     cout<<"Has the character been MISS UNIVERSE ever??" ;
567     int r ; cin>>r ;
568     if(r==1) { f246(ans) ; return ; }
569     if(r==2) { f247(ans) ; return ; }
570     f246(ans) ; f247(ans) ;
571 }
572 void f122( vector<string>&ans )
573 {
574     cout<<"Has the character recently played the role of RANI LAKSHMIBAI??\n"
575     int r ; cin>> r ;
576     if( r==1 ) { f244(ans) ; return ;  }
577     if(r==2)
578     {
579         string s="KANGANA RANAUT" ; ans.push_back(s) ; return  ;
580     }
581     string s="KANGANA RANAUT" ; ans.push_back(s) ; f244(ans) ;
582 }
```

RECURSIVE FUNCTIONS

```
void f140( vector<string>&ans )
{
    cout<<"Is the person famous for his theory of relativity??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        f280(ans) ; return ;
    }
    if(r==2)
    {
        string s="ALBERT EINSTEIN" ; ans.push_back(s) ; return ;
    }
    f280(ans) ;
    string s="ALBERT EINSTEIN" ; ans.push_back(s) ;
}
void f128( vector<string>&ans )
{
    cout<<"Is your character a big fighter of his time??\n" ;
    int r ; cin>>r ;
    if(r==2)
    {
        string s="ALEXANDER" ; ans.push_back(s) ;
        s="ADOLF HITLER" ; ans.push_back(s) ;
        return ;
    }
    if(r==1)
    {
        string s="CHRISTOPHER COLUMBUS" ; ans.push_back(s) ;
        s="GAUTAM BUDDHA" ; ans.push_back(s) ;
        s="NELSON MANDELA" ; ans.push_back(s) ;
        s="ABRAHAM LINCOLN" ; ans.push_back(s) ;
        s="STEVE JOBS" ; ans.push_back(s) ;
        s="DENNIS RITCHIE" ; ans.push_back(s) ;
        return ;
    }
    string s="ALEXANDER" ; ans.push_back(s) ;
    s="ADOLF HITLER" ; ans.push_back(s) ;
    s="NELSON MANDELA" ; ans.push_back(s) ;
    s="DENNIS RITCHIE" ; ans.push_back(s) ;
    s="STEVE JOBS" ; ans.push_back(s) ;
    s="CHRISTOPHER COLUMBUS" ; ans.push_back(s) ;
    s="GAUTAM BUDDHA" ; ans.push_back(s) ;
```

```
void f153( vector<string>&ans )
{
    cout<<"Did he became first president/prime-minister of India??\n";
    int r ; cin>>r ;
    if(r==2)
    {
        string s="MUHAMMAD ALI JINNAH" ; ans.push_back(s) ; return ;
    }
    if(r==1)
    {
        string s="Dr RAJENDRA PRASAD" ; ans.push_back(s) ;
        s="JAWAHARLAL NEHRU" ; ans.push_back(s) ; return ;
    }
    string s="MUHAMMAD ALI JINNAH" ; ans.push_back(s) ;
    s="Dr RAJENDRA PRASAD" ; ans.push_back(s) ;
    s="JAWAHARLAL NEHRU" ; ans.push_back(s) ;
}
void f152( vector<string>&ans )
{
    cout<<"Is the character referred as father of the nation??\n" ;
    int r ; cin>>r ;
    if( r==1 ) { f304(ans) ; return;  }
    if(r==2)
    {
        string s="MAHATMA GANDHI" ; ans.push_back(s) ; return;
    }
    f304(ans) ;
    string s="MAHATMA GANDHI" ; ans.push_back(s) ;
}
```

RECURSIVE FUNCTIONS

```
void f177( vector<string>&ans )
{
    cout<<"Is your character from tech company??\n" ;
    int r; cin>>r ;
    if(r==1)
    {
        string s="AMBANIS" ; ans.push_back(s) ;
        s="TATAS" ; ans.push_back(s) ;
        return ;
    }
    if( r==2 )
    {
        string s="SATYA NADELLA" ; ans.push_back(s) ;
        s="VIJAY SHEKHAR SHARMA" ; ans.push_back(s) ;
        s="SUNDAR PICHAI" ; ans.push_back(s) ;
        return ;
    }
    string s="AMBANIS" ; ans.push_back(s) ;
    s="TATAS" ; ans.push_back(s) ;
    s="SATYA NADELLA" ; ans.push_back(s) ;
    s="VIJAY SHEKHAR SHARMA" ; ans.push_back(s) ;
    s="SUNDAR PICHAI" ; ans.push_back(s) ;
}
void f176( vector<string>&ans )
{
    cout<<"Is the character a singer??\n" ;
    int r ; cin>>r ;
    if( r==1 ) {  return ; }
    if( r==2 )
    {
        string s="ATIF ASLAM" ; ans.push_back(s) ;
        s="ARIJIT SINGH" ; ans.push_back(s) ;
        s="A R RAHMAN" ; ans.push_back(s) ;
        s="SHANKAR MAHADEVAN" ; ans.push_back(s) ;
        s="SONU NIGAM" ; ans.push_back(s) ;
        return ;
    }
        string s="ATIF ASLAM" ; ans.push_back(s) ;
    s="ARIJIT SINGH" ; ans.push_back(s) ;
    s="A R RAHMAN" ; ans.push_back(s) ;
    s="SHANKAR MAHADEVAN" ; ans.push_back(s) ;
    s="SONU NIGAM" ; ans.push_back(s) ;
```

```
void f179( vector<string>&ans )
{
    cout<<"Is your character the current prime minister of INDIA??\n";
    int r ; cin>>r ;
    if( r==1 )
    {
        string s="AMIT SHAH" ; ans.push_back(s) ;
        s="RAJNATH SINGH" ; ans.push_back(s) ;
        return ;
    }
    if(r==2)
    {
        string s="NARENDRA MODI" ; ans.push_back(s) ; return ;
    }
    string s="AMIT SHAH" ; ans.push_back(s) ;
    s="RAJNATH SINGH" ; ans.push_back(s) ;
    s="NARENDRA MODI" ; ans.push_back(s) ;
}
void f178( vector<string>&ans )
{
    cout<<"Is your character a part of congress party??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        string s="ARVIND KEJARIWAL" ; ans.push_back(s) ;
        s="MANISH SISODIYA" ; ans.push_back(s) ;
        return ;
    }
    if(r==2)
    {
        string s="RAHUL GANDHI" ; ans.push_back(s) ; return ;
    }
    string s="ARVIND KEJARIWAL" ; ans.push_back(s) ;
    s="MANISH SISODIYA" ; ans.push_back(s) ;
    s="RAHUL GANDHI" ; ans.push_back(s) ; return ;
}
```

```
void f241(vector<string>&ans  )
{
    cout<<"Does your character play with a bat or a racket??\n" ;
    int r ;cin>>r ;
    if(r==1){ f482(ans) ; return;  }
    if(r==2){ f483(ans) ; return ;  }
    f482(ans) ;  f483(ans) ;
}

void f183( vector<string>&ans )
{
    cout<<"Does he bowl??\n" ;
    int r ; cin>>r ;
    if(r==1){ f366(ans) ; return ; }
    if(r==2) { f367(ans) ; return ;  }
    f366(ans) ;  f367(ans) ;
}
void f182( vector<string>&ans )
{
    cout<<"Is the player better known by its sirname??\n" ;
    int r;cin>>r ;
    if(r==1)
    {
        string s="YUVRAAJ SINGH" ; ans.push_back(s) ;
        s="KAPIL DEV" ; ans.push_back(s) ; return ;
    }
    if(r==2)
    {
        f364(ans) ; return ;
    }
    string s="YUVRAAJ SINGH" ; ans.push_back(s) ;
    s="KAPIL DEV" ; ans.push_back(s) ;
    f364(ans) ; return ;
}
```

```
void f246( vector<string>&ans )
{
    cout<<"Is the character married to a cricketer??\n" ;
    int r; cin>>r;
    if(r==2)
    {
        string s="ANUSHKA SHARMA" ; ans.push_back(s) ; return ;
    }
    if(r==1)
    {
        string s="RANI MUKHERJEE" ; ans.push_back(s) ;
        s="DEEPIKA PADUKONE" ; ans.push_back(s) ;
        s="RAVEENA TANDON" ; ans.push_back(s) ;
        s="MADHURI DIXIT" ; ans.push_back(s) ;
        s="VIDYA BALAN" ; ans.push_back(s) ;
        s="KAREENA KAPOOR" ; ans.push_back(s) ;
        return;
    }
    string s="ANUSHKA SHARMA" ; ans.push_back(s) ;
    s="RANI MUKHERJEE" ; ans.push_back(s) ;
    s="DEEPIKA PADUKONE" ; ans.push_back(s) ;
    s="RAVEENA TANDON" ; ans.push_back(s) ;
    s="MADHURI DIXIT" ; ans.push_back(s) ;
    s="VIDYA BALAN" ; ans.push_back(s) ;
    s="KAREENA KAPOOR" ; ans.push_back(s) ;
}
void f244( vector<string>&ans )
{
    cout<<"Is she seen as a young girl of bollywood??\n" ;
    int r ; cin>>r ;
    if( r==1 ) { f488(ans)  ; return ;  }
    if(r==2)
    {
        string s="ALIA BHATT" ; ans.push_back(s) ;
        s="ANANYA PANDEY" ; ans.push_back(s) ; return ;
    }
    f488(ans)  ;
    string s="ALIA BHATT" ; ans.push_back(s) ;
    s="ANANYA PANDEY" ; ans.push_back(s) ;
}
```

RECURSIVE FUNCTIONS

```
void f280( vector<string>&ans )
{
    cout<<"Is your character famous for invention of electric bulb??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        string s="GALILEO GALILEI" ; ans.push_back(s) ;
        s="FLEMING" ; ans.push_back(s) ;
        s="MICHAEL FARADAY" ; ans.push_back(s) ;
        s="ARCHIMEDES" ; ans.push_back(s) ;
        return;
    }
    if(r==2)
    {
        string s="THOMAS ALVA EDISON" ; ans.push_back(s) ; return;
    }
    string s="GALILEO GALILEI" ; ans.push_back(s) ;
    s="FLEMING" ; ans.push_back(s) ;
    s="MICHAEL FARADAY" ; ans.push_back(s) ;
    s="ARCHIMEDES" ; ans.push_back(s) ;
    s="THOMAS ELVA EDISON" ; ans.push_back(s) ;
}
void f247( vector<string>&ans )
{
    cout<<"Is the character married to an Indian??\n" ;
    int r; cin>>r ;
    if(r==1)
    {
        string s="PRIYANKA CHOPRA" ; ans.push_back(s) ; return ;
    }
    if(r==2)
    {
        string s="AISHWARYA RAI BACHCHAN" ; ans.push_back(s) ; return ;
    }
    string s="PRIYANKA CHOPRA" ; ans.push_back(s) ;
    s="AISHWARYA RAI BACHCHAN" ; ans.push_back(s) ;
}
```

```
void f364( vector<string>&ans )
{
    cout<<"Is there a movie/biopic in bollywood of this character??\n" ;
    int r; cin>>r ;
    if(r==1) { f728(ans) ; return ; }
    if(r==2)
    {
        string s="MAHENDRA SINGH DHONI" ; ans.push_back(s) ;
        s="SACHIN TENDULKAR" ; ans.push_back(s) ; return;
    }
    f728(ans) ;
    string s="MAHENDRA SINGH DHONI" ; ans.push_back(s) ;
    s="SACHIN TENDULKAR" ; ans.push_back(s) ;
}
void f304( vector<string>&ans )
{
    cout<<"Does the suffix \"azad\" come after his name??\n " ;
    int r  ; cin>>r ;
    if(r==2)
    {
        string s="MAULANA ABUL KALAM AZAD" ; ans.push_back(s) ;
        s="CHANDRA SHEKHAR AZAD" ; ans.push_back(s) ;
        return ;
    }
    if(r==1)
    {
        f608(ans) ; return;
    }
    string s="MAULANA ABUL KALAM AZAD" ; ans.push_back(s) ;
    s="CHANDRA SHEKHAR AZAD" ; ans.push_back(s) ;
    f608(ans) ;
}
```

```
void f367( vector<string>&ans )
{
    cout<<"Does he have a brother also who is a cricketer??\n"
    int r ; cin>>r ;
    if(r==1)
    {
        string s="JASPRIT BUMRAH" ; ans.push_back(s) ; return
    }
    if(r==2)
    {
        string s="HARDIK PANDYA" ; ans.push_back(s) ;
        s="KRUNAL PANDYA" ; ans.push_back(s) ;
        return ;
    }
    string s="JASPRIT BUMRAH" ; ans.push_back(s) ;
    s="HARDIK PANDYA" ; ans.push_back(s) ;
    s="KRUNAL PANDYA" ; ans.push_back(s) ;
}
void f366( vector<string>&ans )
{
    cout<<"Is the character better known as GABBAR??\n" ;
    int r; cin>>r ;
    if(r==1) { f732(ans) ; return ; }
    if(r==2)
    {
        string s="SHIKHAR DHAWAN" ; ans.push_back(s) ; return
    }
    f732(ans) ;
    string s="SHIKHAR DHAWAN" ; ans.push_back(s) ;
}
```

RECURSIVE FUNCTIONS

```
void f483( vector<string>&ans )
{
    cout<<"Is your character a badminton player??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        string s="SANIA MIRZA" ; ans.push_back(s) ;
        s="MITHALI RAJ" ; ans.push_back(s) ;
        return ;
    }
    if( r==2 )
    {
        string s="SAINA NEHWAL" ; ans.push_back(s) ;
        s="PV SINDHU" ; ans.push_back(s) ;
        return ;
    }
    string s="SANIA MIRZA" ; ans.push_back(s) ;
    s="MITHALI RAJ" ; ans.push_back(s) ;
    s="SAINA NEHWAL" ; ans.push_back(s) ;
    s="PV SINDHU" ; ans.push_back(s) ;
}
void f482( vector<string>&ans )
{
    cout<<"Is your character a wrestling player??\n" ;
    int r ; cin>>r ;
    if(r==1)
    {
        string s="MARY KOM" ; ans.push_back(s) ;
        s="DIPA KARMAKAR" ; ans.push_back(s) ;
        return ;
    }
    if( r==2 )
    {
        string s="GITA PHOGAT" ; ans.push_back(s) ;
        s="BABITA PHOGAT" ; ans.push_back(s) ;
        return ;
    }
    string s="MARY KOM" ; ans.push_back(s) ;
    s="DIPA KARMAKAR" ; ans.push_back(s) ;
    s="GITA PHOGAT" ; ans.push_back(s) ;
    s="BABITA PHOGAT" ; ans.push_back(s) ;
}
```

```
void f608( vector<string>&ans )
{
    cout<<"Is STATUE OF UNITY made in your character's honour??\n" ;
    int r ; cin>>r ;
    if(r==2)
    {
        string s="SARDAR VALLABH BHAI PATEL" ; ans.push_back(s) ; return ;
    }
    if(r==1)
    {
        f1216(ans) ; return;
    }
    string s="SARDAR VALLABH BHAI PATEL" ; ans.push_back(s) ; f1216(ans) ;
}
void f488( vector<string>&ans )
{
    cout<<"Has she been in any controversy at some point of time with ranbir kapoor??\n" ;
    int r; cin>>r;
    if(r==2)
    {
        string s="KATRINA KAIF" ; ans.push_back(s) ; return ;
    }
    if(r==1) { f976(ans) ; return ; }
    f976(ans) ; string s="KATRINA KAIF" ; ans.push_back(s) ;
}
```

```
void f732( vector<string>&ans )
{
    cout<<"Is the character current Indian captain??\n";
    int r; cin>>r ;
    if( r==1 )
    {
        string s="ROHIT SHARMA" ; ans.push_back(s) ; return ;
    }
    if( r==2 )
    {
        string s="VIRAT KOHLI" ; ans.push_back(s) ; return ;
    }
    string s="ROHIT SHARMA" ; ans.push_back(s) ;
    s="VIRAT KOHLI" ; ans.push_back(s) ;
}
void f728( vector<string>&ans )
{
    cout<<"Was the character part of ICC CRICKET WORLD CUP 2011 winning team??\n" ;
    int r; cin>>r ;
    if(r==1)
    {
        string s="SUNIL GAVASKAR" ; return ;
    }
    if( r==2 )
    {
        string s="GAUTAM GAMBHIR" ; ans.push_back(s) ;
        s="VIRENDRA SEHWAG" ; ans.push_back(s) ;
        s="SURESH RAINA" ; ans.push_back(s) ;  return ;
    }
    string s="SUNIL GAVASKAR" ;
    s="GAUTAM GAMBHIR" ; ans.push_back(s) ;
    s="VIRENDRA SEHWAG" ; ans.push_back(s) ;
    s="SURESH RAINA" ; ans.push_back(s) ;
}
```

RECURSIVE FUNCTIONS

```
112 void f1216( vector<string>&ans )
113 {
114     cout<<"Is your character the writer of national anthem of india??\n" ;
115     int r ; cin>>r ;
116     if( r==2 )
117     {
118         string s="RABINDRANATH TAGORE" ; ans.push_back(s) ; return;
119     }
120     if(r==1) { f2432(ans) ; return; }
121     string s="RABINDRANATH TAGORE" ; ans.push_back(s) ; f2432(ans) ;
122 }
123 void f976( vector<string>&ans )
124 {
125     cout<<"Has the character graduated from an engineering college??\n" ;
126     int r ; cin>>r ;
127     if(r==1)
128     {
129         string s="SHRADDHA KAPOOR" ; ans.push_back(s) ; return ;
130     }
131     if(r==2)
132     {
133         string s="KRITI SANON" ; ans.push_back(s) ; return ;
134     }
135     string s="SHRADDHA KAPOOR" ; ans.push_back(s) ;
136     s="KRITI SANON" ; ans.push_back(s) ;
137 }
```

```
89  using namespace std;
90  void f2432(  vector<string>&ans )
91  {
92      cout<<"Is your character a saint??\n" ;
93      int r ; cin>>r ;
94      if(r==2)
95      {
96          string s="SWAMI VIVEKANANDA" ; ans.push_back(s) ; return ;
97      }
98      if(r==1)
99      {
100         string s="MANGAL PANDEY" ; ans.push_back(s) ;
101         s="VIR SAVARKAR" ; ans.push_back(s) ;
102         s="Dr B R AMBEDKAR" ; ans.push_back(s) ;
103         s="BATUKESHWAR DUTT" ; ans.push_back(s) ;
104         return;
105     }
106     string s="SWAMI VIVEKANANDA" ; ans.push_back(s) ;
107     s="VIR SAVARKAR" ; ans.push_back(s) ;
108     s="MANGAL PANDEY" ; ans.push_back(s) ;
109     s="Dr B R AMBEDKAR" ; ans.push_back(s) ;
110     s="BATUKESHWAR DUTT" ; ans.push_back(s) ;
111 }
```

RECURSIVE FUNCTIONS

SCREENSHOTS OF SOME OUTPUTS

```
                WELCOME TO THE PROGRAM:  YOU ARE FOUND



here, the program will ask you simple questions regarding the basic properties of the person and hence will try to figure out the
    character going in in your mind


HOPE WE GUESS THE CHARACTER CORRECTLY




So here we begin
Press 1 to continue
1


Basic Rules:
. You will be asked simple YES/NO questions
. If you dont know the answer, reply with 0
. If your answer is NO, reply with 1
. Finally, if your answer is YES, reply with 2

. After a sequence of questions, we shall reach to the character you wished to search for and hence provide other info of that
    character too

So here we begin by the first question
```

COMMON SET OF RULES WHICH ARE PRINTED EVERYTIME THE CODE IS RUN

FIRST EXAMPLE: I THOUGHT OF VIRAT KOHLI

```
So here we begin by the first question
Is your character a female??
1
Is your character alive??
2
Is your character INDIAN??
2
 Is your character an actor??
 1
 Is your character a sportsman??
 2
Is the character a cricketer??
2
Does the character still play international cricket??
2
Does he bowl??
1
Is the character better known as GABBAR??
1
Is the character current Indian captain??
2



(thinking)
.
.
(still thinking)
.
.
(compiling our results)


(compilation done successfully)
OHH YESS, WE GOT OUR ANSWER

BASED ON YOUR GIVEN ANSWERS, THESE ARE THE CHARACTERS YOU PROBABLY WOULD HAVE THOUGHT OF:

1)  VIRAT KOHLI
```

SECOND EXAMPLE: I THOUGHT OF ALBERT EINSTEIN

```
So here we begin by the first question
Is your character a female??
1
Is your character alive??
1
Is your character INDIAN??
1
Is your character a scientist/mathematician/inventor or from any such research oriented profession??
2
Has your character contributed something in the fields of physics??
2
Is the character famous for suggesting the model of an atom (protons,neutrons,eletrons)??
1
Is the character famous for the 3 laws of motion??
1
Is the person famous for his theory of relativity??
2



(thinking)
.
.
(still thinking)
.
.
(compiling our results)


(compilation done successfully)
OHH YESS, WE GOT OUR ANSWER

BASED ON YOUR GIVEN ANSWERS, THESE ARE THE CHARACTERS YOU PROBABLY WOULD HAVE THOUGHT OF:

1)  ALBERT EINSTEIN
```

THIRD EXAMPLE: I THOUGHT OF QUEEN OF JHANSI

```
So here we begin by the first question
Is your character a female??
2
Is your character alive??
1
Is your character INDIAN??
2
 Is your character associated with British-Indian struggle??
 2
Has the character been the prime minister of India??
1
Is the character a well known queen of a territory??
2



(thinking)
.
.
(still thinking)
.
.
(compiling our results)


(compilation done successfully)
OHH YESS, WE GOT OUR ANSWER

BASED ON YOUR GIVEN ANSWERS, THESE ARE THE CHARACTERS YOU PROBABLY WOULD HAVE THOUGHT OF:

1)  RANI LAKSHMIBAI
```

FINAL EXAMPLE: I THOUGHT OF ANUSHKA SHARMA

```
So here we begin by the first question
Is your character a female??
2
Is your character alive??
2
Is your character INDIAN??
2
Is the character a politician??
1
Is the character an actress??
2
Is the character married??
2
Has the character been MISS UNIVERSE ever??
1
Is the character married to a cricketer??
2



(thinking)
.
.
(still thinking)
.
.
(compiling our results)


(compilation done successfully)
OHH YESS, WE GOT OUR ANSWER

BASED ON YOUR GIVEN ANSWERS, THESE ARE THE CHARACTERS YOU PROBABLY WOULD HAVE THOUGHT OF:

1)  ANUSHKA SHARMA
```

FAQ's

WHY TAKE RECURSIVE FUNCTIONS TO BUILD AND IMPLEMENT THE TREE WHEN YOU CAN DO IT USING "STRUCT" BY CREATING NODES???

1) Time complexity of both remains same. But space complexity differs greatly. In our case, space complexity is O(log n) which is used by the recursive stacks, whereas in creating and declaring nodes, space complexity becomes O(n) as whatsoever be the flow of control, all n nodes will be created.
2) If someone wants to add extra features to just some parts of the code, then its quite easy and handy in our case, as simply those functions can be editted , whereas it is not possible in second approach. Either we change the whole structure of the Node and make changes with every entry in the database, else it isn't possible to do so.
3) The questions which are getting printed at each level are not required to be stored in string, simply outputting it using recursive functions save good amount of memory

WHAT IF THERE ARE CONTRADICTORY QUESTIONS OR THE USER IS NOT 100% SURE OF THE ANSWER

We have tried our best to avoid any such contradictory questions. Questions are kept as simple as possible

And still if user isn’t aware of any answer, then an additional feature is added wherein user CAN say that he isn’t aware of the answer and still can reach his destination.

RESULTS

Each and every day, we found out new data for our database when we consulted different people to run out code on their query. Slowly and gradually, we improved upon our pass:fail ratio and hence we were now in a much better position than earlier. This practice continued till the last day of the project and at the end, we got quite desirable outcomes and now our code worked correctly on most of the queries.

FUTURE DEVELOPMENTS

This program that was being presented was simply guessing the name of the famous personality the user thought at that time and found it out. Keeping the overall skelatal same and just changing the questions and the database, we can implement the same program to serve immense purposes in day-to-day lives in different areas.

SOME QUITE IMPORTANT APPLICATIONS OF THIS PROGRAM ARE :-

LAW AND SECURITY:

This program can be very beneficial when it comes to maintenance of law and security. This code can be updated to store information about various criminals, which are being fetched by the police. Finally, this program can then help in finding out previous criminal records of that person, or even simply recognising whether this person has earlier visited the jail by any other fake name or not. Even for general use also, if someone finds a person suspicious, he can always refer to this program, search for physical traits and get to know whether that suspicious person has a criminal record or not.

MAINTAINING PROPER PHONEBOOK:-

A normal person can also use this program to keep a record of his/her contacts. He can enter the details of that person here and at any point of time, search for that person through any of the details he mentioned earlier.

eg: if user just remembers that he met an acquaintance in a particular place , or he remembers that they both worked together at a particular place, then only by inputting this data, user can see all the people who match with the inputted information.

FOR FUN PURPOSES:

User can always use this program to challenge the program of how many people this program can recognise correctly and for how many people, the program doesn't have the required database. This is indeed a good way also to learn about more of such iconic characters which the user might not be aware of.

Other than these applications, this project can be efficiently used wherever TRACKING is required. Be it in some security purpose or for domestic purpose too.

CONTRIBUTIONS

- IMPLEMENTING THE CODE: AVIKAL
- DEBUGGING: AVIKAL/ABHIJIT
- SUGGESTING NEW CHARACTERS: AVIKAL/ABHIJIT
- POWER POINT PRESENTATION: AVIKAL/ABHIJIT
- ADDITIONAL RESEARCH WORK AND IMPROVEMENTS: AVIKAL

REFERENCES

- https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm
- https://www.geeksforgeeks.org/binary-tree-data-structure/
- https://www.geeksforgeeks.org/recursion/
- https://www.youtube.com/watch?v=KEEKn7Me-ms
- https://en.wikipedia.org/wiki/The_100:_A_Ranking_of_the_Most_Influential_Persons_in_History
- https://en.wikipedia.org/w/index.php?search=Forbes+list+of+The+World%27s...&title=Special%3ASearch&fulltext=1&ns0=1
- https://www.newidea.com.au/who-is-the-most-famous-person-in-the-world
- https://www.biographyonline.net/people/famous-100.html
- https://bloggernow.in/list-of-famous-personalities/
- https://leverageedu.com/blog/great-personalities/