\* INTRODUCTION:

    ATm in c is based on a concept of managing an account personally.From this project user can check total balance,Mini Statement,Money Withdrawal,Paying Bills,Deposit Money and generate new pin easily and the user can save time without going to banks.

\* Process of Creating this project:

   \* Created a new repository account in github

   \* clone this repository account to Visual studio code using some commands.

   \* Created a new folder for creating this project.

   \* During Compilation we need to follow 4 steps:

      \* Preprocessing :

        \* Tool:Cpp

         \* It takes input of .c file and it check if there are any commands such as multiline commands and it removes it.

         \* Substitute the content of header files

         \* Preprocessers will be substituted in its place like #define #define #if etc

         \* It generates pure c code without macros and comments.

         \* gcc -E <filename>

         \* Output generated in .i file

      \* Compilation:

        \* Tool:gcc

          \* It can check syntax,proper naming conventions,semi colons,variable mapping,semantics

          \* It generates an assembly language code

          \* If you want to stop during compilation we need to use gcc-S<filename>command and generates .s file

      \* Assembling:

        \* Tool: as

          \* Converts assembly code to machine code i.e object code

          \* gcc -c <c source code file>

          \* It generates .o file

      \* Linking:

        \* Tool:ld

          \* Adreess of external function like library or other files is linked during runtime and compile time

   \* Create files like header files,Make files,test case files etc

   \* we need to do unit testing by manually following AAA method

      \* Arrange: Initialize the piece of the application to be used

      \* Act: Invoke the units and capture the results

      \* Assert: Observe and retrospect the behaviour

\* There are 2 methods for Unit Testing

    \* Manual

    \* Automation

\* I have done unit testing by using Unity Throw Switch

\* Learning Outcomes:

  \* I have learned about how to to create any project using multifile and makefile.

  \* I have learned about Unit Testing

  \* I have learned about how to make structural and behavioural diagrams

  \* I have learned about different linux commands

\* Softwares Used:

  \* Visual Studio Code

  \* WSL-Ubuntu

  \* git