

An Assignment on:

**Lab project on Data structures**

**Course code:** CSE 106

**Course title**: Data structures Lab

**Project**: **Pseudo Intelligent chat bot**

**Submission date**: 17 Dec 2019

**Submitted to: Submitted by:**

**MD ANSARUL ISLAM SHARTAZ YEASAR**

Lecturer, **191002130**

Dept. of CSE 193DJ

CONTENTS

**Chapter 01**

* 1. **Introduction**
  2. **Requirement & Analysis**

**Chapter 02**

**2.1 Beginning**

**2.2 Learn**

**2.3 Listen**

**2.4 Decide**

**2.5 Reply**

**Chapter 03**

**3.1 Run project**

**3.2 Test project**

**CHAPTER 01**

* 1. **INTRODUCTION**

Friday is a pseudo intelligent chat bot that is made for entertainment purpose. This is a Data structures based project. This includes some data structures operations and a large 3 dimensional array. As it is a chat bot, it takes input from user and processes it some in logical way to show the output. In the 3D array which is defined as ‘Base’ there are a lot of potential inputs and compatible reply/outputs. So it acts like Artificial intelligent but it is not. So it’s only goal is entertainment. In code, there are some user defined functions beside build in functions. Additionally, this project can be increased/upgraded to any limit any time adding more data in the main array (Base). As a result the more data we put in it, the more intelligently it acts.

* 1. **REQUIREMENT AND ANALYSIS**

To make this project complete we mostly used a C++ compiler ‘Code blocks’.

It required a lots of variables and arrays. There are 13 user defined functions divided in three sections

Major user defined functions

Minor user defined functions &

Extra user defined functions

Considering human behavior, an intelligent being reacts to certain situations through some conventional but complex ways. Namely, when someone talk to them, they first hear then they tries to understand, compares with experiences, and then decides what to react.

In this chat bot the code flows exactly the same way. The code let the compiler have a string input aka message from user. Then it reads the message. And

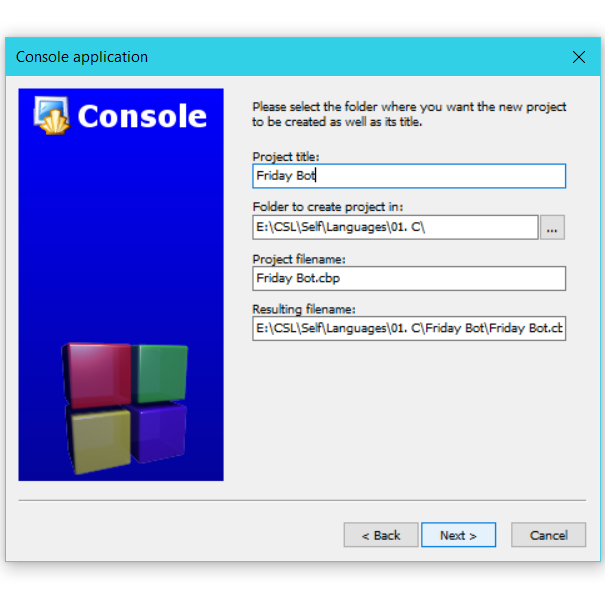
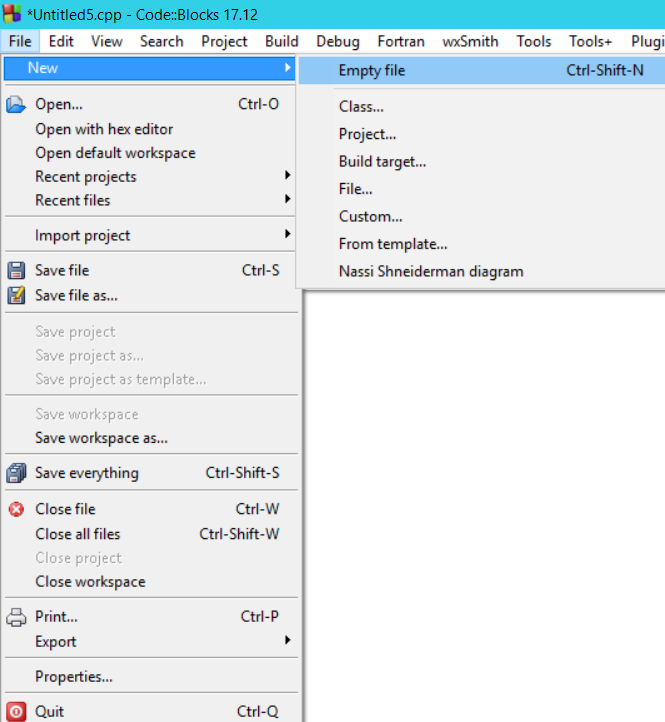
then compares the message with elements of a large array to find the most quotable answers. After going through these steps it decides the best answer and shows it on the screen.

Also it saves the message given by user to understand what the user is talking about and to choose best answers for next messages. Which makes it pseudo or fake intelligent.

**CHAPTER 02**

**2.1 Beginning**

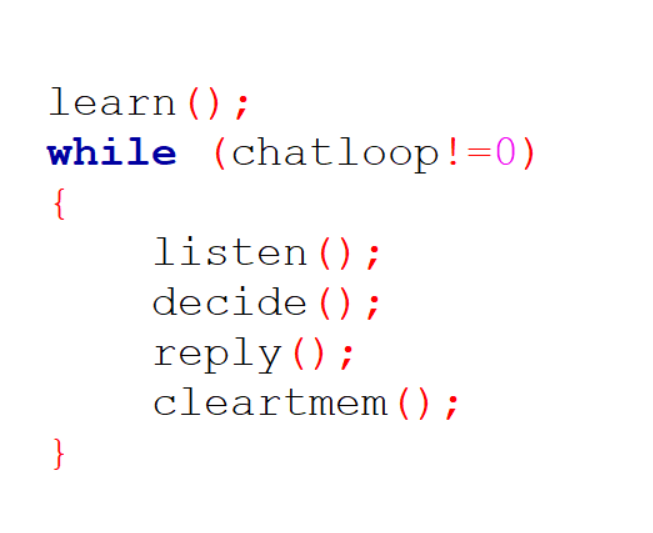
First of all we created a new project using code blocks.

****

**2.2 Learn**

We wrote a code to show a welcome message and instructions. Then, by pressing enter the main screen loads. Literally the code begins from main function. Which decorates the screen and runs a loop which calls the very user major defined functions as follows.

* **Learn**
* **Listen**
* **Decide**
* **Reply**

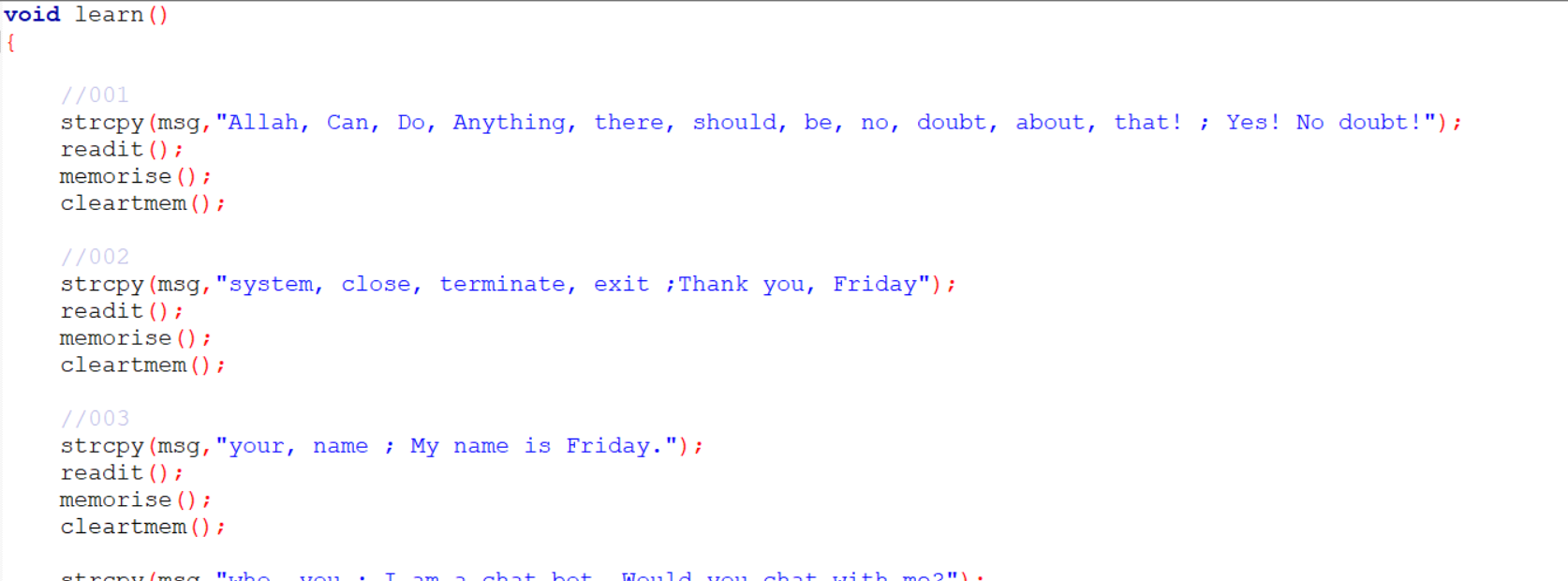


**2.3 CONVERTER PART**

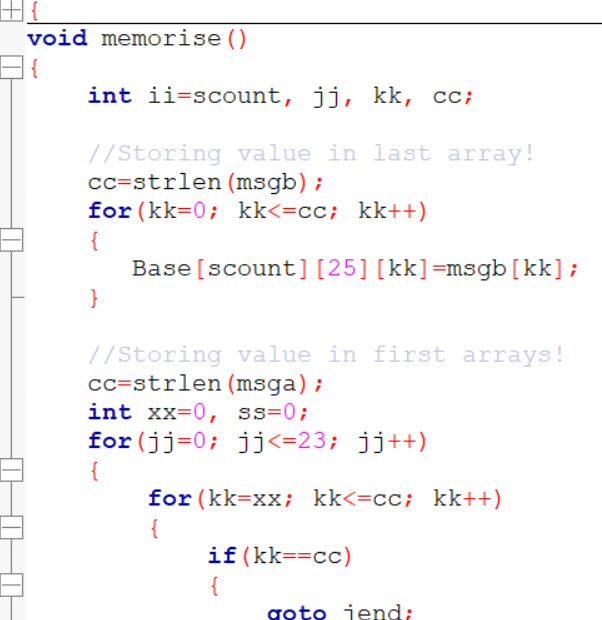
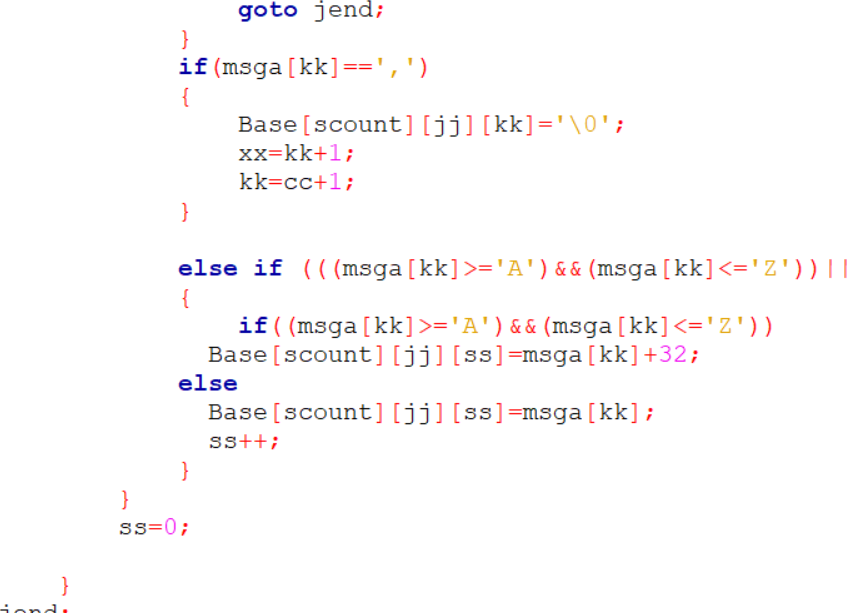
After learn function is called, flow jumps into learn() which contains a lot of potential inputs and compatible reply/outputs. It also calls another two different functions to store the learn items perfectly.

It calls read it () and memories it () function. Which is a special type of compiler to convert learn items. Read it function breaks the message into two parts called msg a and msg b to save them separately in base.

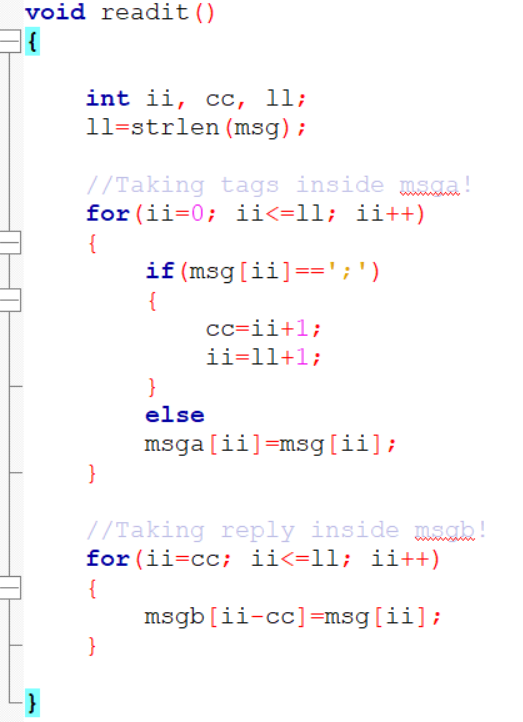
Memories function stores the broken parts in base.



In memorise function:

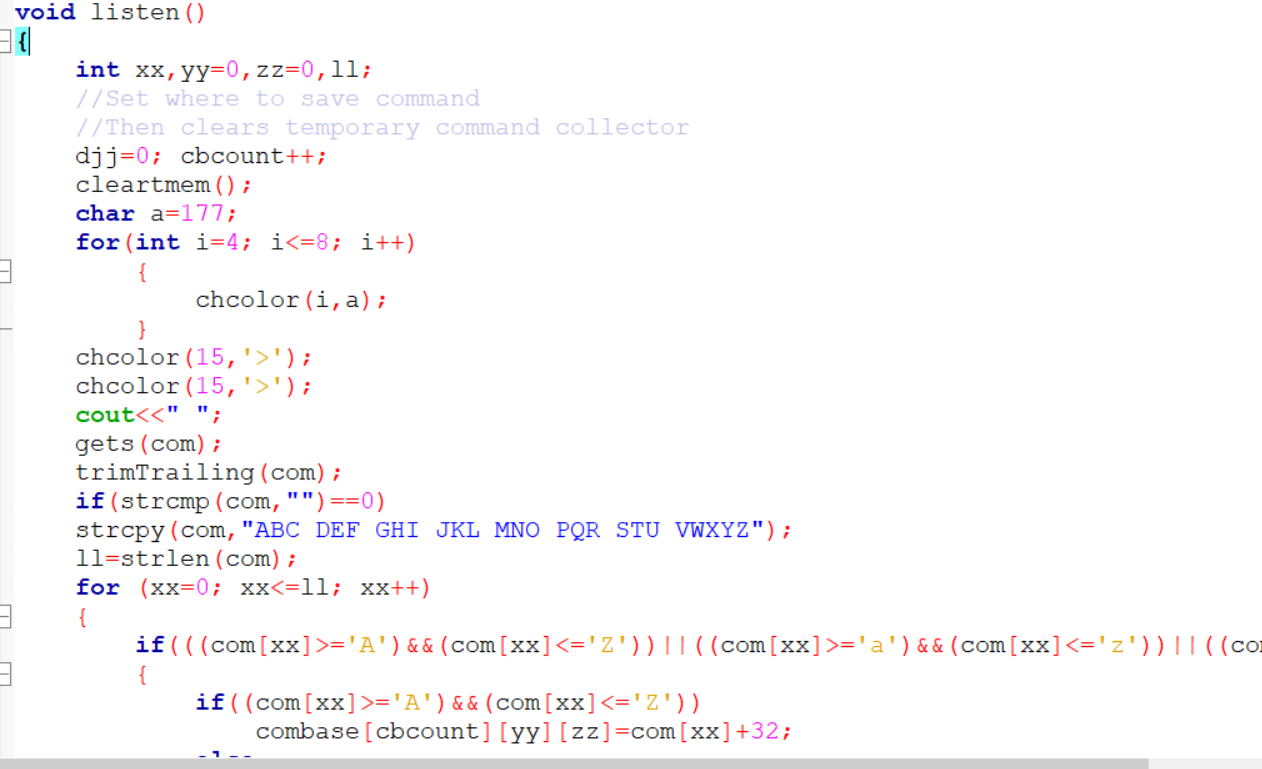
In read it function:



**2.3 Listen**

In next step while listen function is called, the function takes input from user.

This function saves the message in a message base.

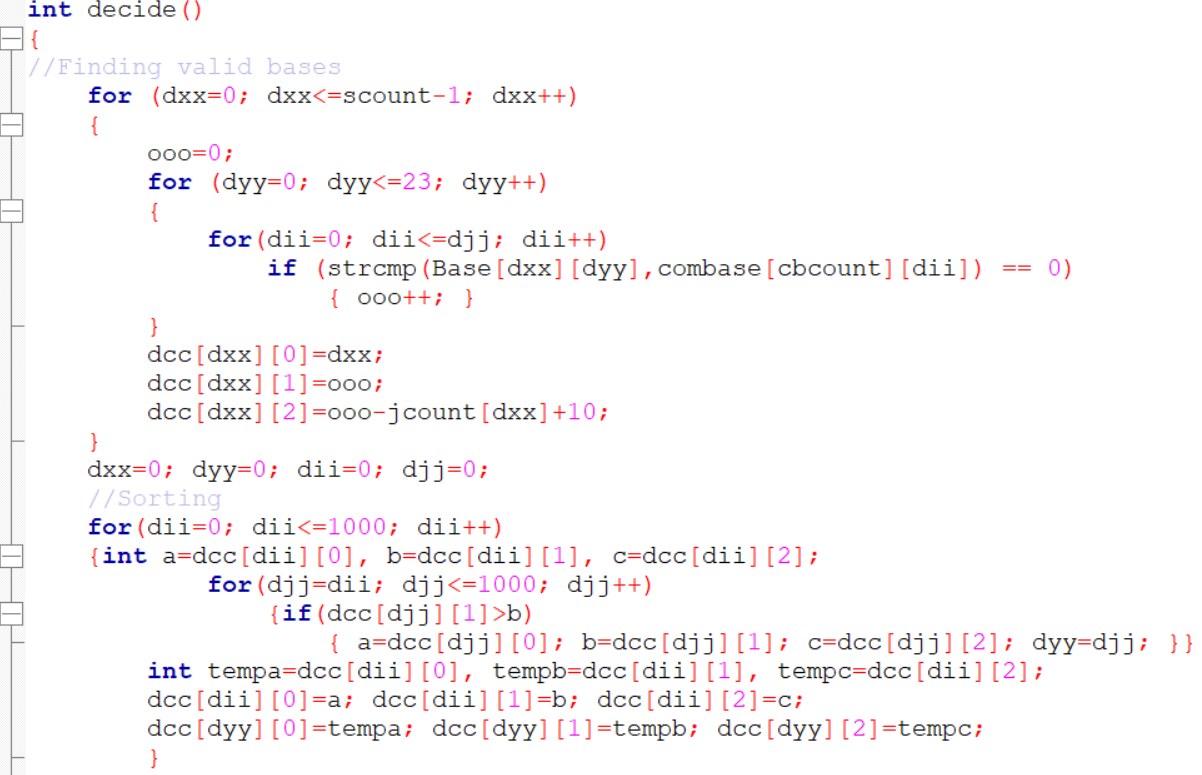
****

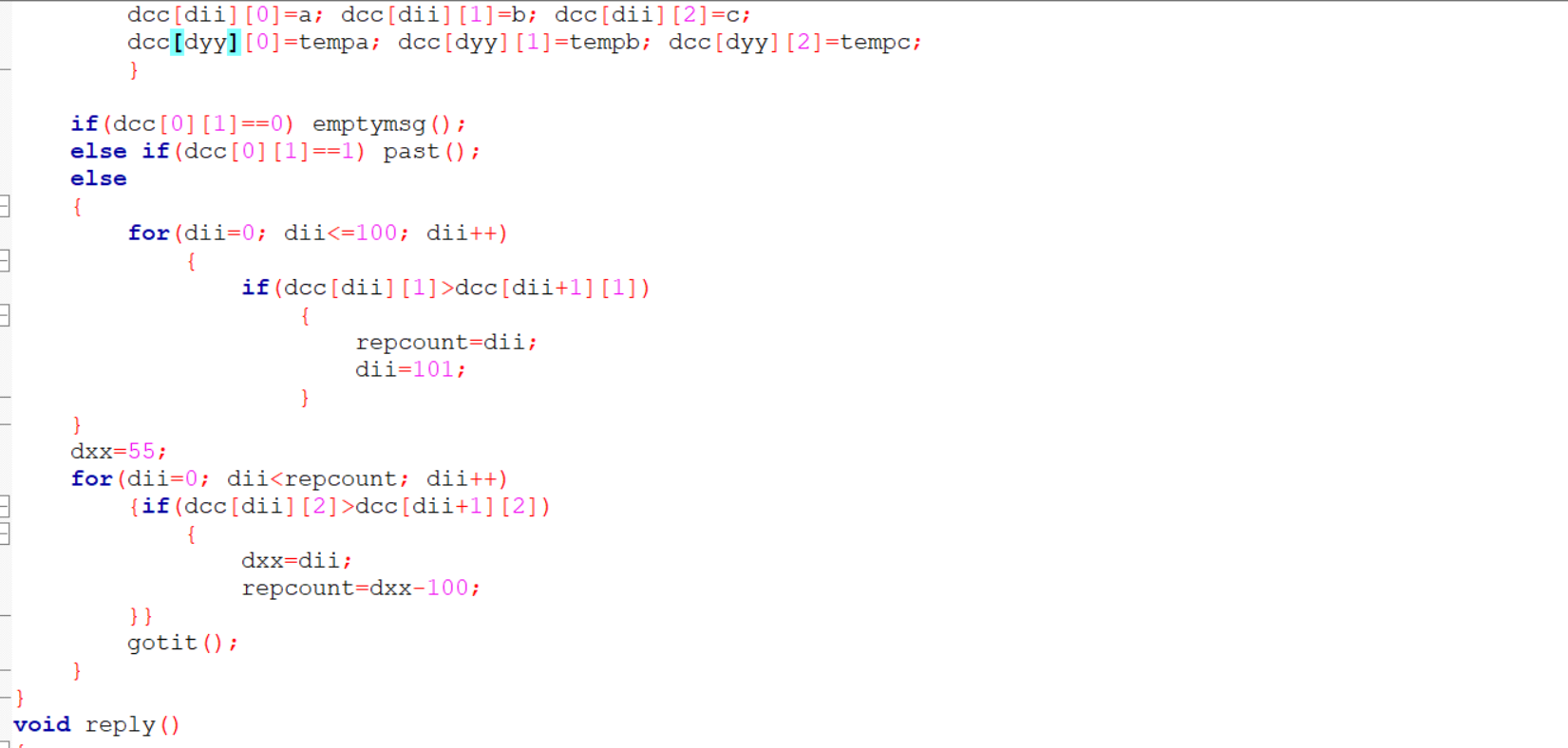
**2.4 Decide**

Main function calls decide function after works of learn and listen are done. This function decides what message to reply comparing the message given by user with all base messages. Also this function calls three different functions to make the decision as wise as possible. Those are:

* Empty function: Reacts to unknown and no input message.
* Got it function: Reacts to a full and valid message.
* Past function: Makes a weak or unclear message clear to answer intelligently.

**Decide function:**

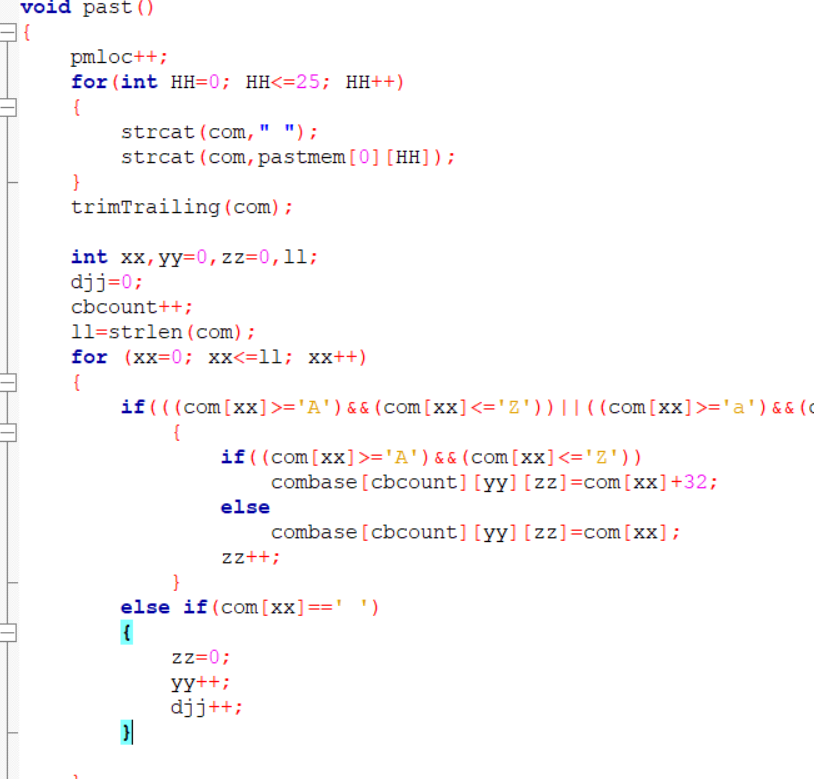




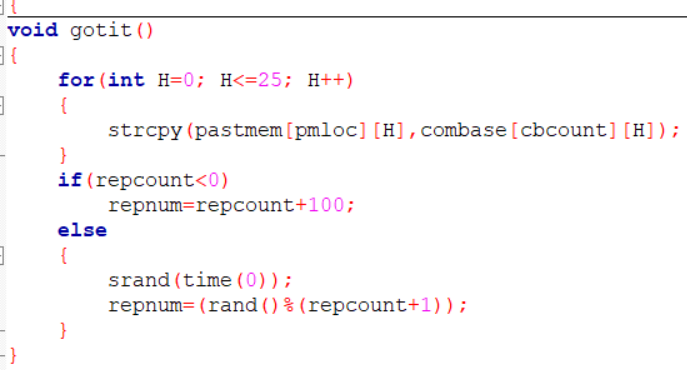
**Empty message it function:**



**Past function:**

****

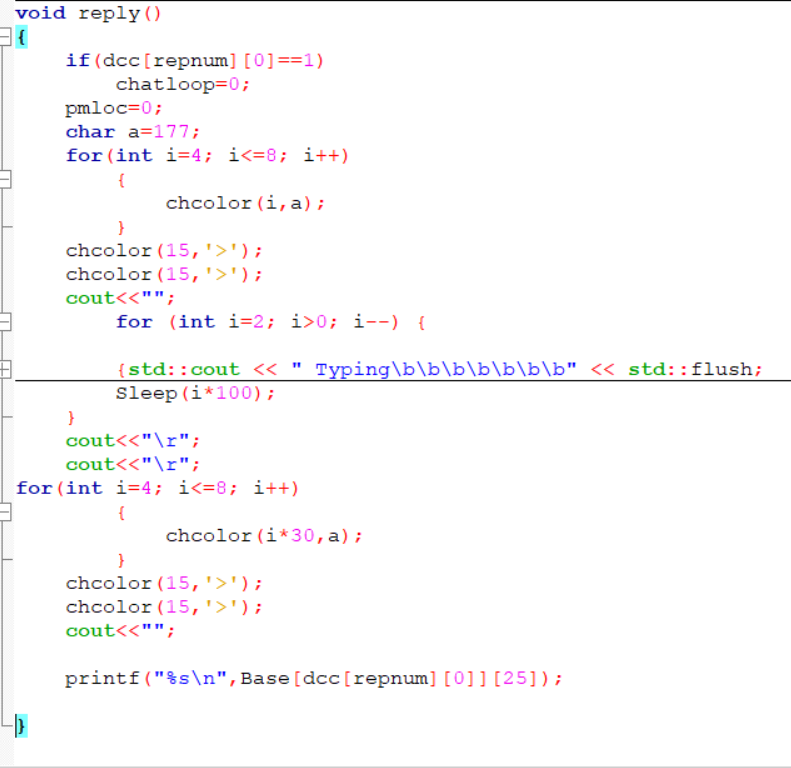
**Got it function:**

****

**2.5 Reply**

After all these steps reply functions comes to decorate the screen once again and show the most quotable reply. It takes help of some another extra functions to decorate the screen.

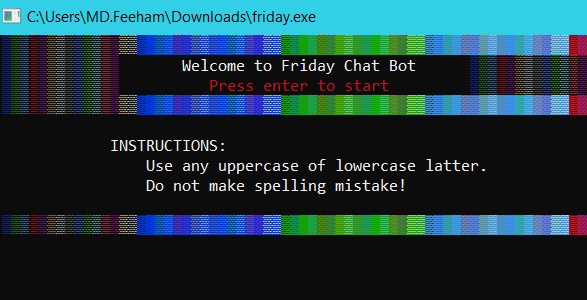
**Reply function:**

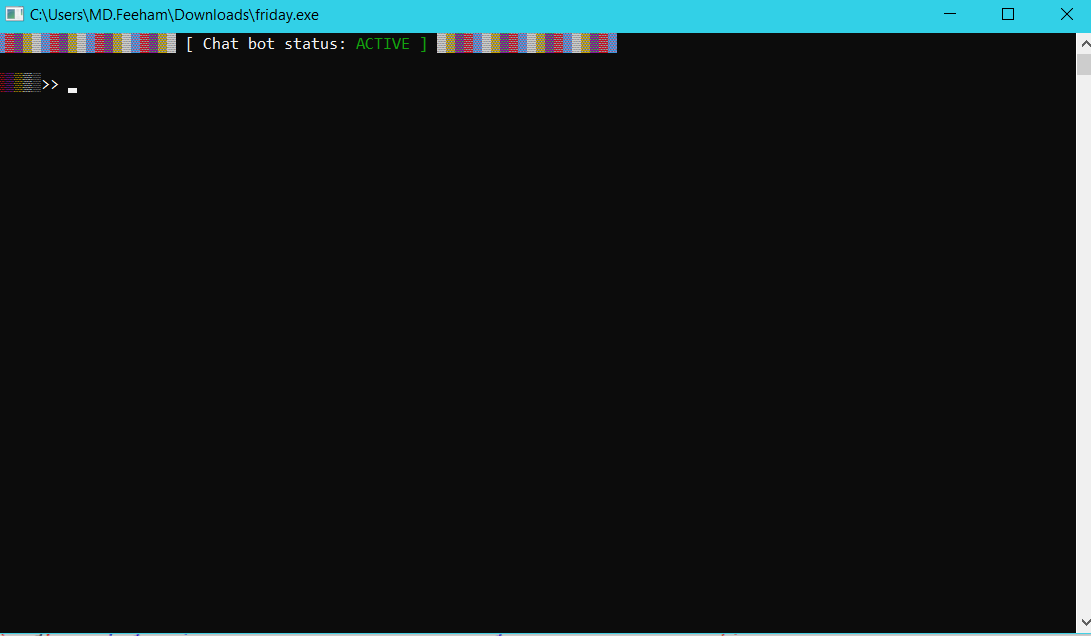
****

**Additionally, there are some extra functions to help the implementation.**

**3.1 RUN PROJECT**

After successful compilation, the code runs perfectly.

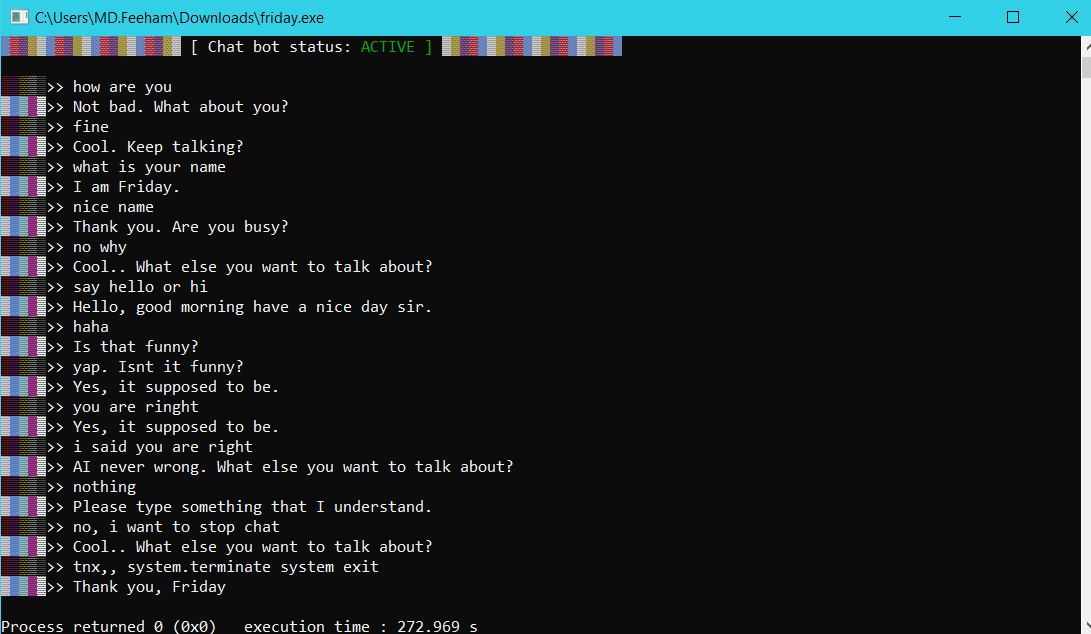




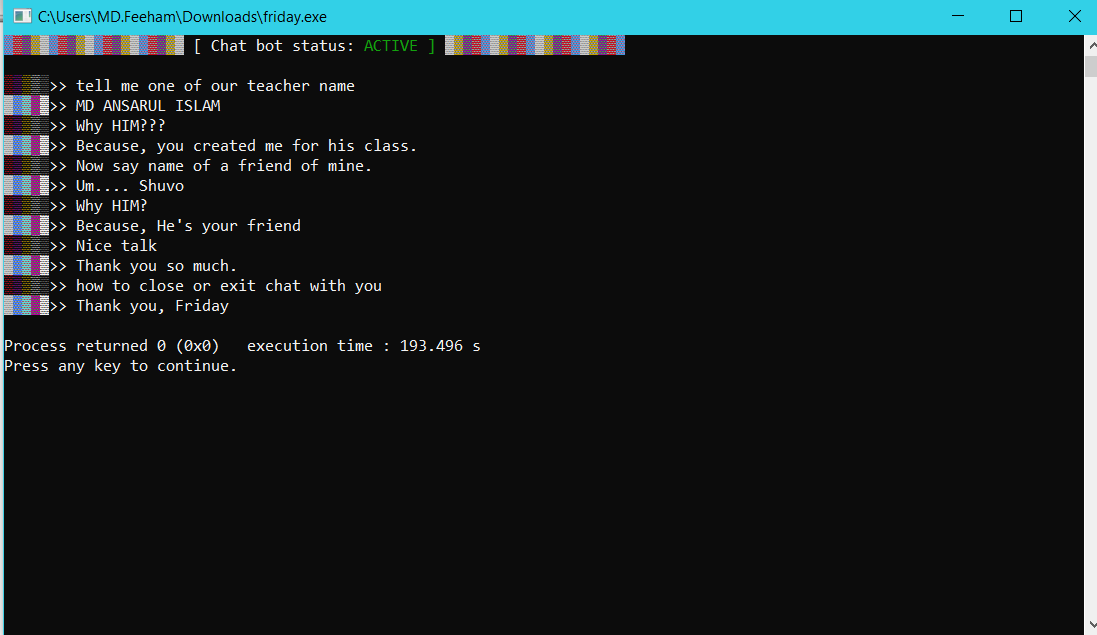
**3.2 Test project**

**Project on different test cases:**

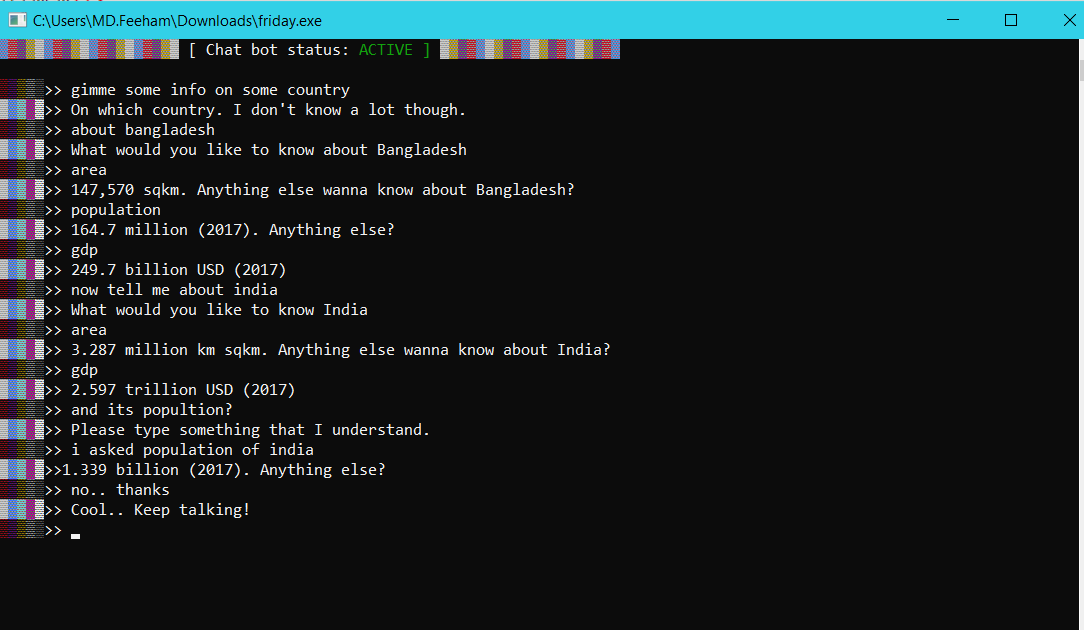
**#1**



**#2**



**#3**

****

**[THE END]**