**1. Introduction about the system**

**I. Industry selected:** The industry selected is charity industry.

**II. Company background**

The company is a non-profit foundation center named Hopes Connected Foundation. The goal of the organization is to utilize the growing trend of ICT to enhance various kinds of charitable activities. This foundation was founded by Mr. Kiri Kaniathran who receives a well reputation in charity career and be well-known as the market leader in Malaysia’s IT industry. He proposed that the disadvantaged group should not be forgotten in the great wave of globalization.

The establishment of this foundation is one of his efforts to deliver the tangible benefits of the advance technology to the vulnerable group. Mr. Kiri Kaniathran would like to bring public together involved in the charity movement by exploiting the propagation properties of the contemporary technology. He believes that the combination of the technology and charity industry will bring a new wave of energy into the traditional philanthropy industry. By providing a convenient channel, people will have a higher willingness to dedicate themselves into the charitable activities.

On the other hands, with the nature of the software system, almost all the transactions involved supports audit trail and this enhance the transparency of the charity activities which in turns consolidate people’s confidence towards the charity organization. Mr. Kiri Kaniathran would also like to provide an open software platform for public to rate and comment the mainstream charity organizations and activities. This not only serves as urging and supervision to the charitable group but also a pool of opinions sourced by public which will generate great ideas to further beef up current charitable effort.

**III. System name:** The system name is Hopes Connected Charity Integrated System

**IV. Functions of the Program**

- Login

- Display Menu

- Rate and Comment Charity Organizations

- Donation (Cash)

- Donation (Goods)

- Join Volunteer

**V. Formulas used**

**-** One-Off Donation = no. of notes \* (RM100 or RM50 or RM20 or RM10 or RM5 or RM1)

- Actual amount received for Monthly Donation = Donation – (Donation\*5%)

- Donation by second hand dealer = Current Goods Value\*1.05

- Rating (in Stars) = Round (Total Rating / no. of raters)

**VI. Assumptions**

**-** 5% of the monthly donation is charged or extracted out to deal with the daily expenses that maintain the operation of the foundation.

- For donated goods which are too obsolete to be used anymore, a kind second hand dealer named Chee Keong Second Hand Sdn. Bhd will receive those goods and donate 5% of the current goods value to the foundation.

**VII. Flow chart**

Start

True

False

False

Less than 3 times ?

Exit

4

3

2

1

Join Volunteer

Donation (Goods)

Donation (Cash)

Rate and Comment

Choice?

Display Menu

True

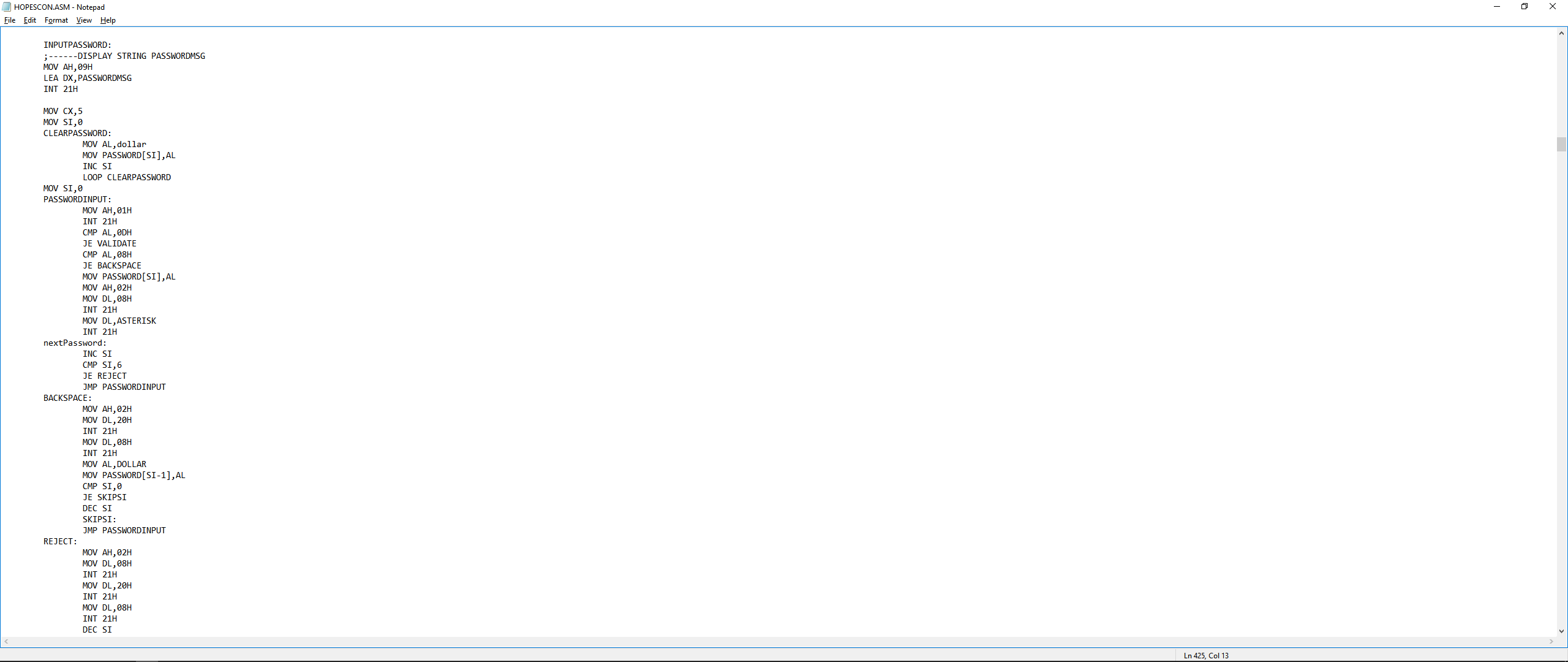
Password ?

Accept Menu Choice

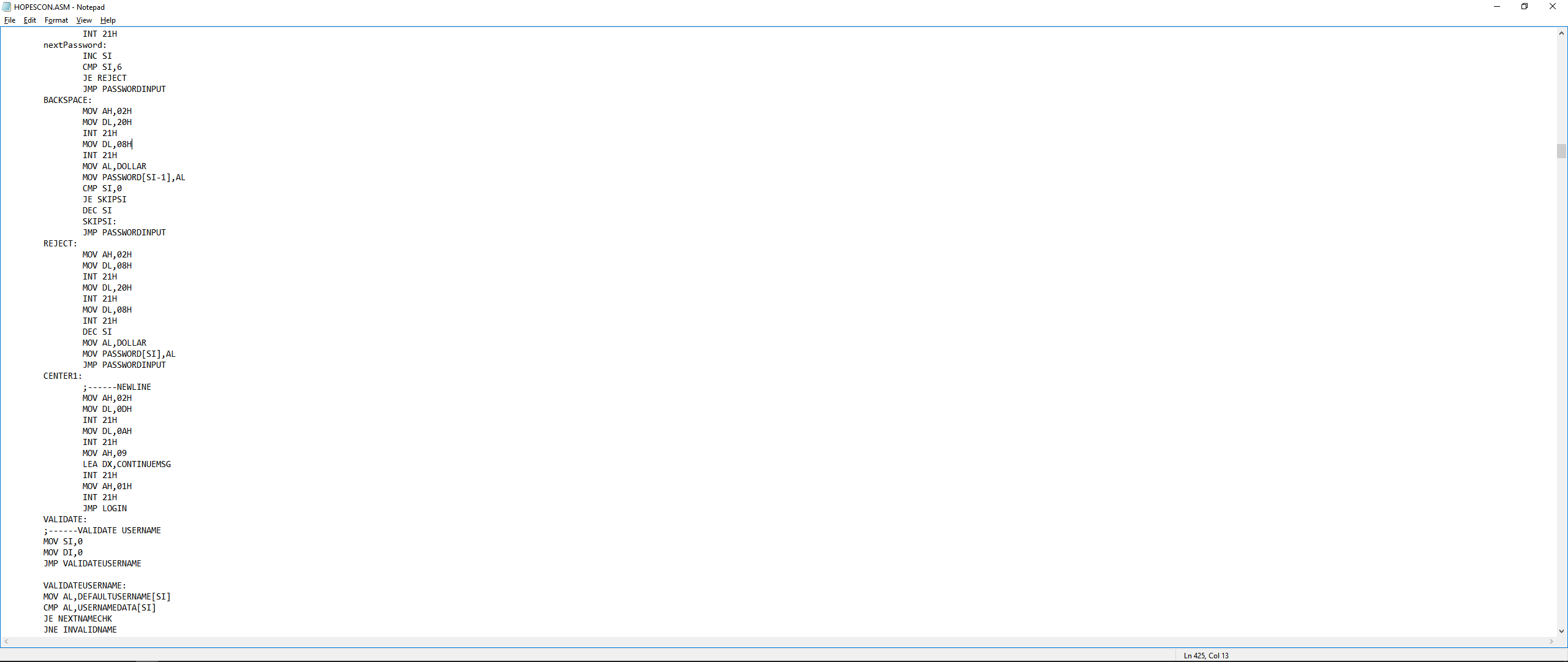
**2. Coding & Logic**

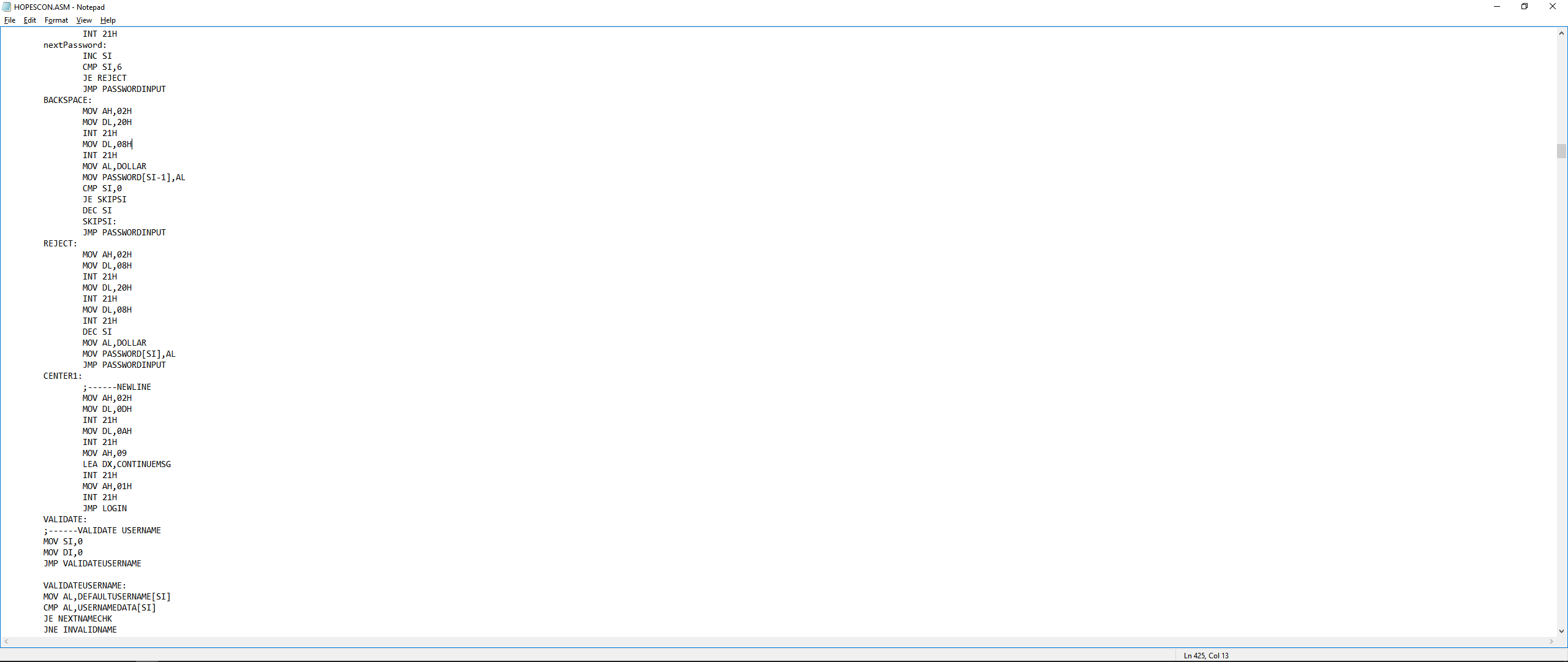
|  |
| --- |
| **Logic Flow**  **INPUTPASSWORD:**  These three lines are used for displaying password input message  **CLEARPASSWORD:**  This CLEARPASSWORD loop is used for clearing the password entered before to reassign ‘$’ characters to the PASSWORD array.  **PASSWORDINPUT:**  01H is used instead of 07H to accept input of password.  Input stored in AL will then compare with 0DH  (enter key/carriage return), 08H (backspace character). If 0DH is entered then the program directly jump to validate the username and password so user can have freedom on input how many characters for password as long as it is less than five. If 08H is entered then the program will jump to backspace label then perform the self-defined backspace function.  While other inputs will be moved from AL to the respective location in the PASSWORD array. After this, program will call 02H function with INT 21H interrupt to display 08H backspace character which will move the cursor one place to the front then call INT 21H again to display the self-defined ASTERISK character (‘\*’) to replace the user input character. With this way, user input will be hidden by the ASTERISK character without affecting the correct input process.  **nextPassword:**  Then system will increment SI by one then compare with 6. If it is equal then it will jump to  self-defined reject function which used to ensure the user input is not more than five. If SI is less than 6 then the program will jump back to PASSWORDINPUT to accept the next character input for PASSWORD array. |

**I. Extremely User Friendly Login Password Input**



|  |
| --- |
| **BACKSPACE:**  Once the user input backspace character, the cursor will move one place to the front then this BACKSPACE function will call 02H function with INT 21H interrupt to display 20H space character to replace/cover up the particular character one place in the front then display 08H backspace character to move cursor one place to the front one more than times so the next input will be displayed on the right position. However, all these are just done for displaying purpose but the input is still being stored in PASSWORD array. Thus, this function will use AL register as medium to move self-defined DOLLAR character (‘$’) to the PASSWORD array at [SI-1] position which store the input character that user want to delete. Then SI will be decremented by one to make sure next input from user will be stored in the right position of  PASSWORD array. In addition, to prevent the condition that SI is being decremented until less than zero so the SI will be compared with 0 before it is being decremented. If it is 0 then it will straight jump back to the PASSWORDINPUT to accept the next input as the first character stored in PASSWORD array.  **REJECT:**  This function will be jumped when the input from the user is more than five. Then this function will call 02H function with INT 21H interrupt to display 08H backspace character then display 20H then display 08H again. So this is basically repeat the same logic as backspace function which serves as a system automatic backspace function when the input characters are more than five. No matter how many characters user try to type after 5 characters have been input, system will automatically delete every of them. |





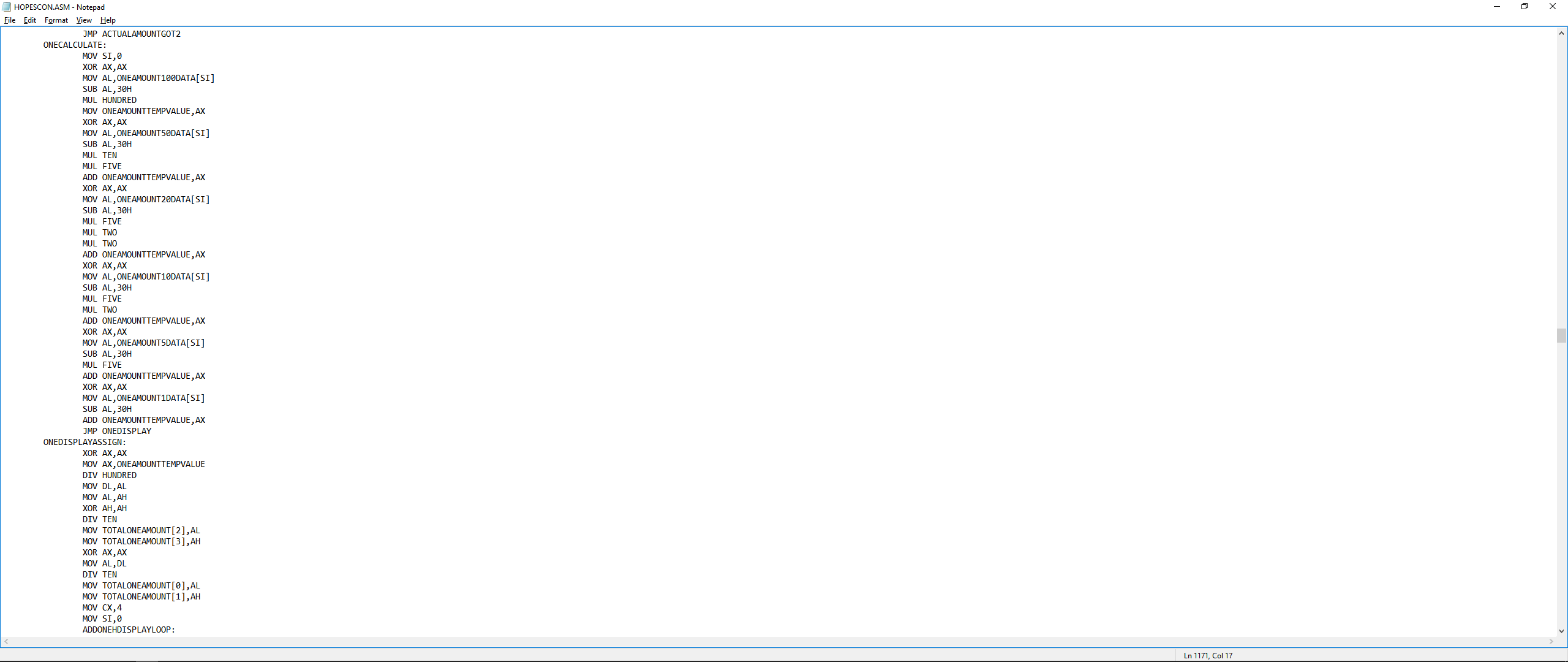
Every of these self-defined functions made the user password input process very friendly and intuitive. Using 01H and ASTERISK character enable user to see clearly how many characters that they have typed without sacrificing the security purpose. The combination of BACKSPACE function and REJECT function can greatly reduce the user input error. This is very important for a charity software system which is seldom used by any regular people who may not clearly and precisely remember the password they set for the account in this system.

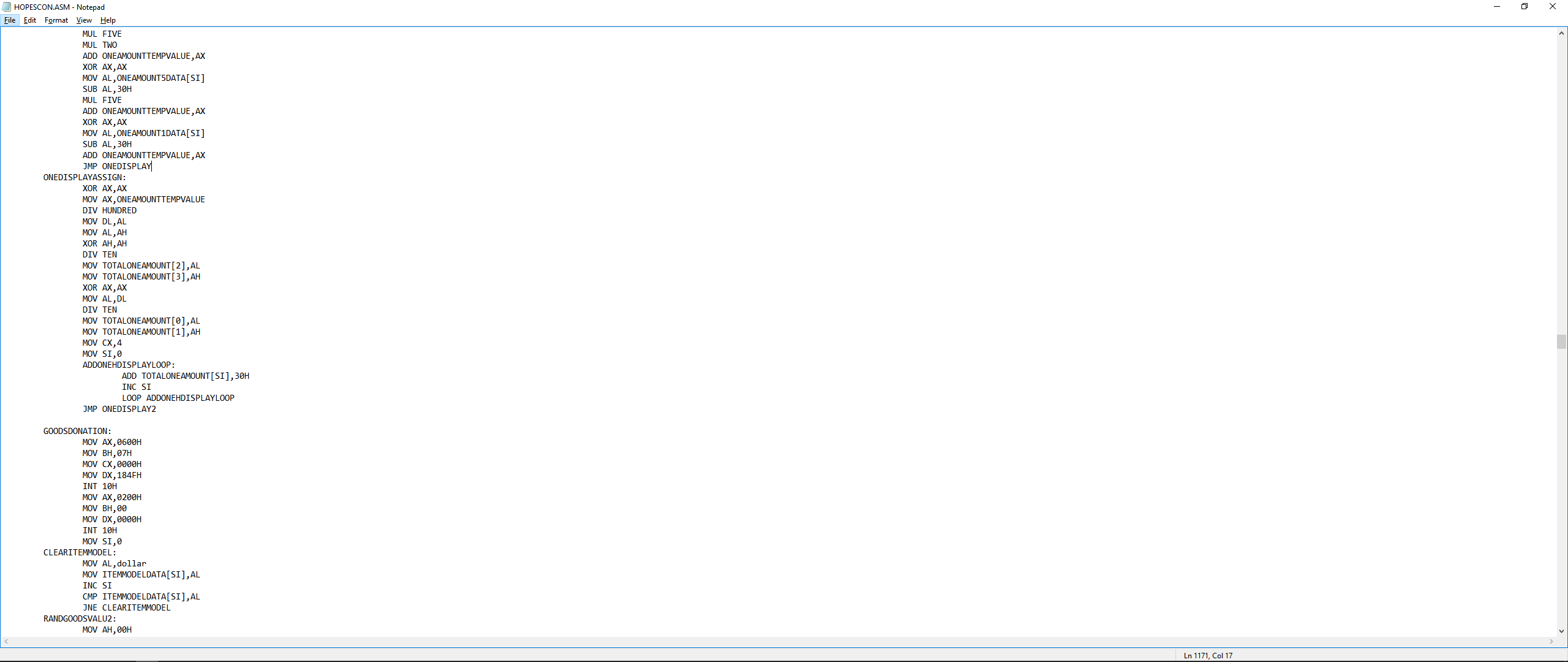
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ONECALCULATE:**  This function is basically adding each individual bank note after multiplying their respective quantities with their faces values into ONEAMOUNTTEMPVALUE will store 4 bytes data.  For RM100,  ONEAMOUNT100DATA[0] which store the quantity for RM100 notes as ASCII code will be moved to AL register then subtract by 30H to get the respective decimal value. This value is then multiplied by 100 and moved to ONEAMOUNTTEMPVALUE. The same logic applied to the rest of the bank notes as this table. (The rest of the value is added to ONEAMOUNTTEMPVALUE instead of moving to get the total.)   |  |  | | --- | --- | | BANKNOTE | MULTIPLY | | RM100 | HUNDRED (100) | | RM50 | TEN(10) AND FIVE (5) | | RM20 | FIVE(5) AND TWO(2) AND TWO(2) | | RM10 | FIVE(5) AND TWO(2) | | RM5 | FIVE(5) | | RM1 | - |   After sum up the value, then we need to display it. ONEAMOUNTTEMPVALUE is first divided by HUNDRED so the quotient (thousands digit and hundreds digit) will be stored in AL and the remainder (tenth value) will be stored in AH. The quotient is moved from AL to DL first then move remainder in AH to the AL. XOR AH, AH is used to clear the AH so the AX only left the remainder/tenth value. Then AX is divided by TEN to separate the ones digit and tens digit into remainder (AH) and quotient (AL) respectively then store them into third and second places in TOTALONEAMOUNT array respectively. Same things goes with thousands digit and hundreds digit after clearing the AX and moving them from DL to AL but stored into the first and second places in TOTALONEAMOUNT array. |

**II. Four Digits and Two Decimal Places Calculation**

**Four Digits Calculation**

**(Sum up One-Off Donation)**



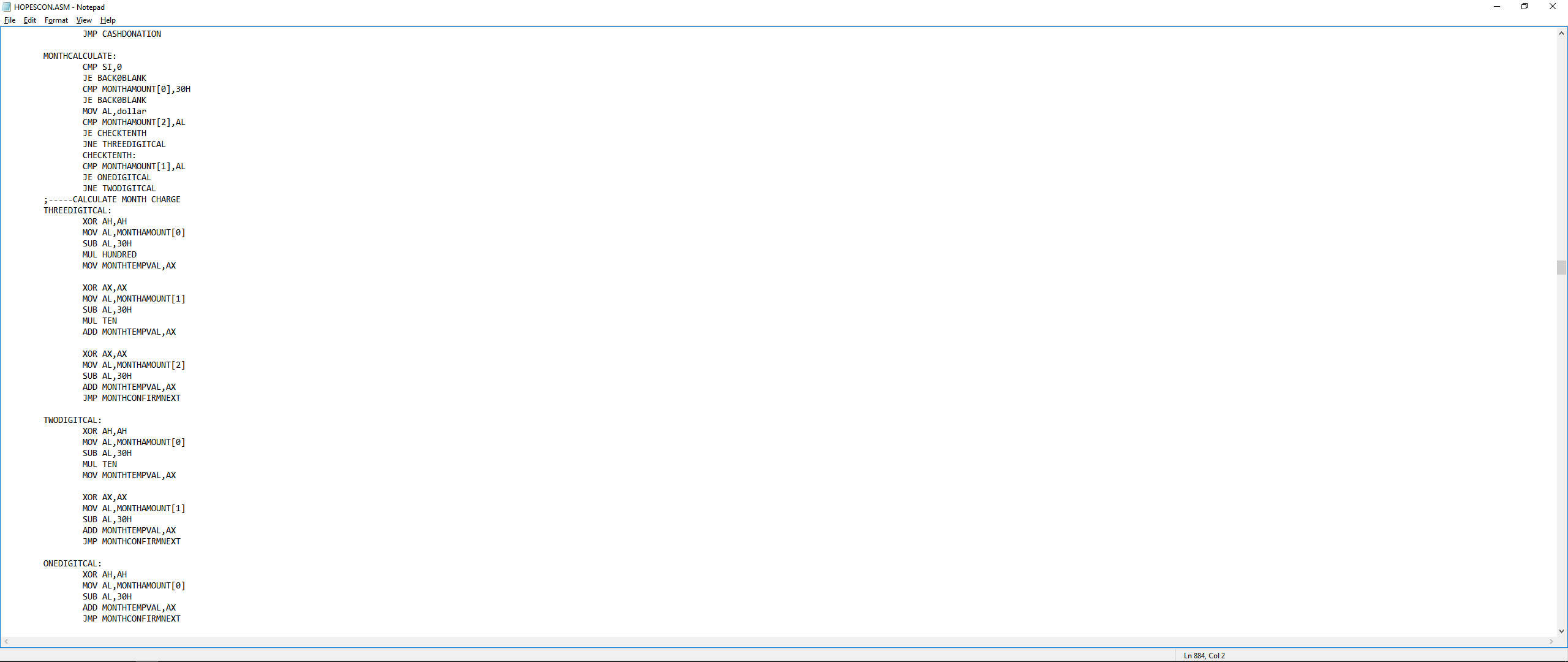


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **TOTALONEAMOUNT array :1674 (Max value in this program)** | | | |
| **Position** | **0** | **1** | **2** | **3** |
| **Value** | **Thousands digit** | **Hundreds digit** | **Tens digit** | **Ones digit** |
| **Example :** | **1** | **6** | **7** | **4** |

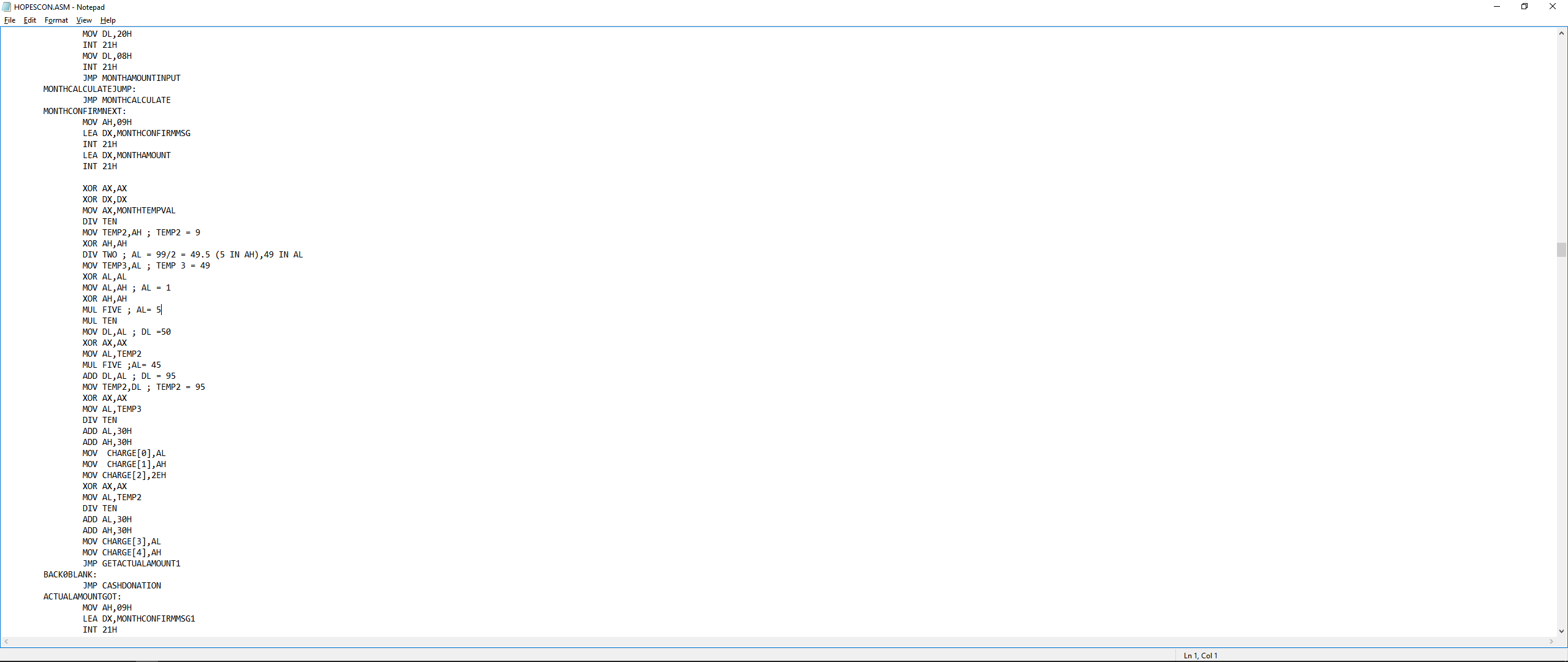
|  |
| --- |
| **MONTHCALCULATE:**  First four lines is too check whether user input is blank or zero such that system will jump to BACK0BLANK which use JMP to jump back to CASHDONATION menu as these two inputs are regarded as back option input. Then, the MONTHAMOUNT[2] and MONTHAMOUNT[1] are then compared with DOLLAR(‘$’) to check whether user entered one, two or three digits value. Then the program will jump to the respective THREEDIGITCAL, TWODIGITCAL or ONEDIGITCAL to make sure the correct amount of value to being moved and added into MONTHTEMPVAL. (Max is RM999, user may enter 9 or 99 or 999 so the MONTHAMOUNT[0] may needed to multiply different values.)  **THREEDIGITCAL:**  MONTHAMOUNT[0] is moved to AL which will be subtracted by 30H to get the exact decimal value then multiply HUNDRED since this is the first digit in a three digits value.Then move the value to MONTHTEMPVAL.  MONTHAMOUNT[1] is moved to AL which will be subtracted by 30H to get the exact decimal value then multiply TEN since this is the second digit in a three digits value. Then add the value to MONTHTEMPVAL.  MONTHAMOUNT[2] is moved to AL which will be subtracted by 30H to get the exact decimal value then add the value to MONTHTEMPVAL since it is the third digit in a three digits value so no multiplication is needed.  **TWODIGITCAL:**  MONTHAMOUNT[0] is moved to AL which will be subtracted by 30H to get the exact decimal value then multiply TEN since this is the first digit in a two digits value.Then move the value to MONTHTEMPVAL.  MONTHAMOUNT[1] is moved to AL which will be subtracted by 30H to get the exact decimal value then add the value to MONTHTEMPVAL since it is the second digit in a three digits value so no multiplication is needed.  **ONEDIGITCAL:**  MONTHAMOUNT[0] is moved to AL which will be subtracted by 30H to get the exact decimal value then add the value to MONTHTEMPVAL since it is the first digit in a three digits value so no multiplication is needed.All these three functions will then jump to MONTHCONFIRMNEXT. |

**Two Decimal Places Calculation**

**(Calculate 5% charges to monthly donation)**

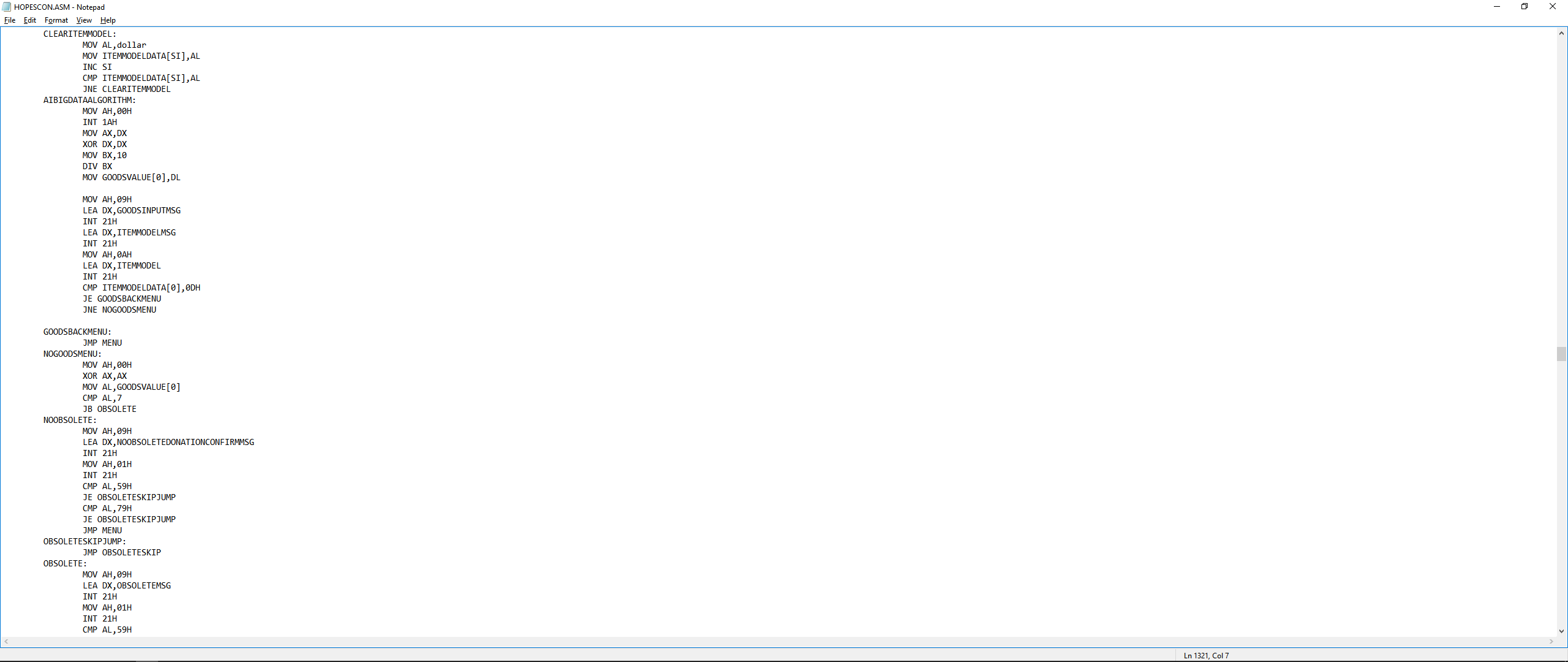


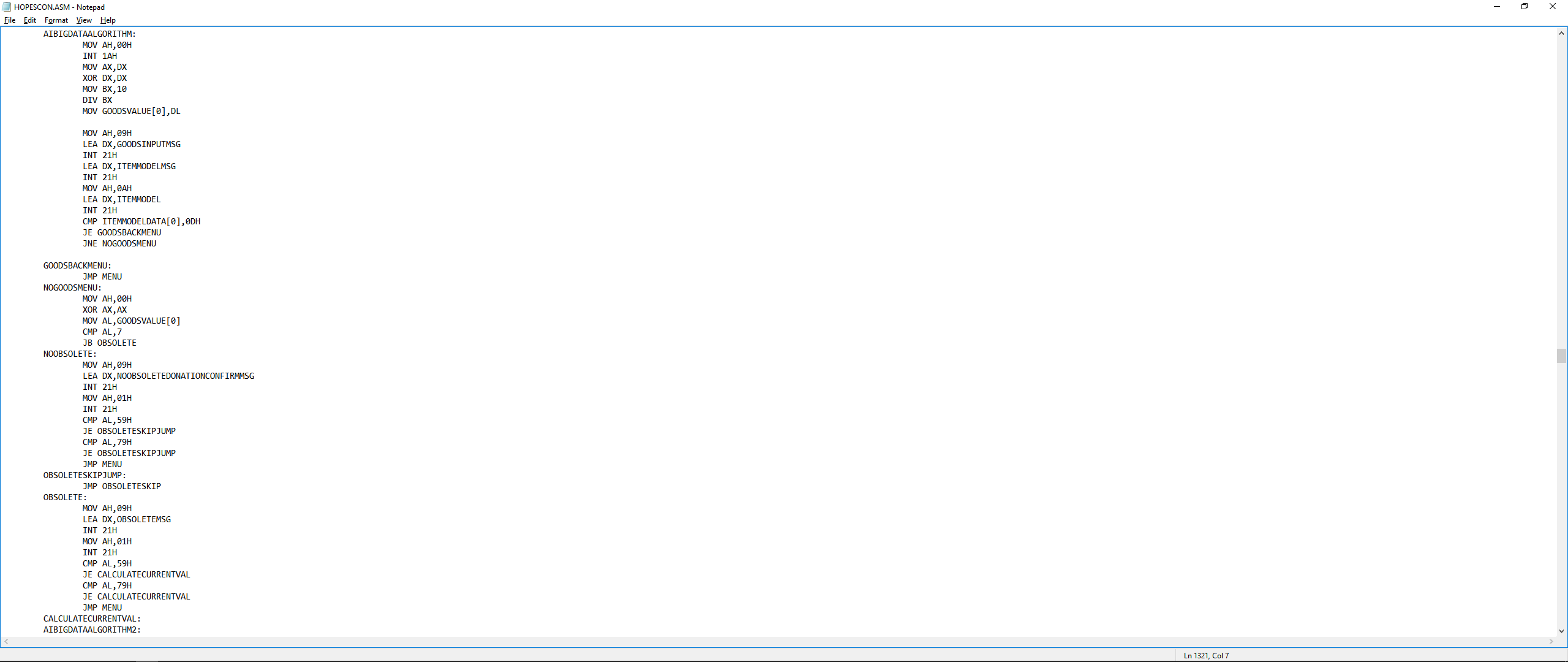
|  |
| --- |
| **MONTHCONFIRMNEXT:**  First five lines are used to display the string for monthly donation confirm message and month amount that stored the input monthly amount from user.  Let’s take value of 999 as example:  Then AX and DX is cleared by XOR to prepare for the proceeding calculations.  MONTHTEMPVAL is first moved to AX then AX is divided by TEN to extract the first digit value into AH and then stored in TEMP2. Then XOR AH, AH to clear the AH, so AX will left only the quotient in AL part. AX is then divided by TWO to get the 5% charge value from the monthly amount.  (Divided by TEN then TWO is equivalent to 0.05)  However, because of the division mechanism in assembly language so we will lose the value for the 0.05 of the third digit so we need to add it back. Since that value is a decimal point value so we need extract the decimal point value from the previous result which is 0.5 or remainder 1 stored in AH. Hence the quotient part (integer value) is moved to TEMP3 from AL before we clear the AL and move AH to AL then clear the AH. AX which is now hold the remainder 1 are multiplied by FIVE then TEN to get the exact decimal point value which is 50. The value is moved from AL to DL first before we clear the AX and move TEMP2 which store the first digit into AL. AX is then multiplied by FIVE to get the exact value in decimal point (9x0.05 = 0.45 so 45 is the exact value in decimal point) then add AL to DL to get the correct decimal point value. (45+50=95). AX is then cleared by using XOR.  For storing as a string that displayed as a two decimal places value, CHARGE array is used.  TEMP3 which store the integer part will be divided by TEN to get the first digit (quotient stored in AL) and second digit (remainder stored in AH), then each of them is added 30H to represent respective ASCII value and stored into CHARGE[0] and CHARGE[1] respectively. CHARGE[2] will store 2EH which is “.” character to separate integer part and decimal point part.  TEMP2 which store the decimal number part will be divided by TEN to get the first digit (quotient stored in AL) and second digit (remainder stored in AH), then each of them is added 30H to represent respective ASCII value and stored into CHARGE[3] and CHARGE[4] respectively. |

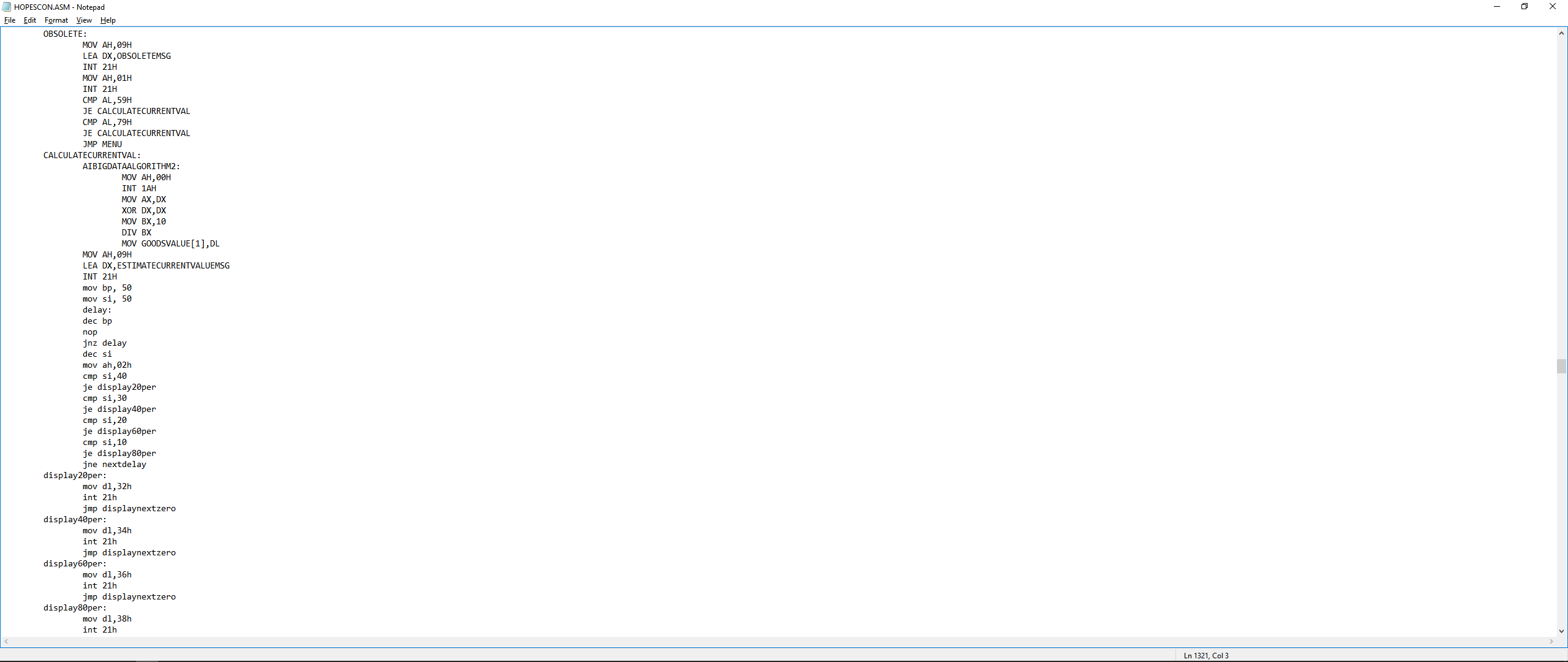


**III. Determine Obsolete or not and Estimating Current Value of Item in Goods Donation**

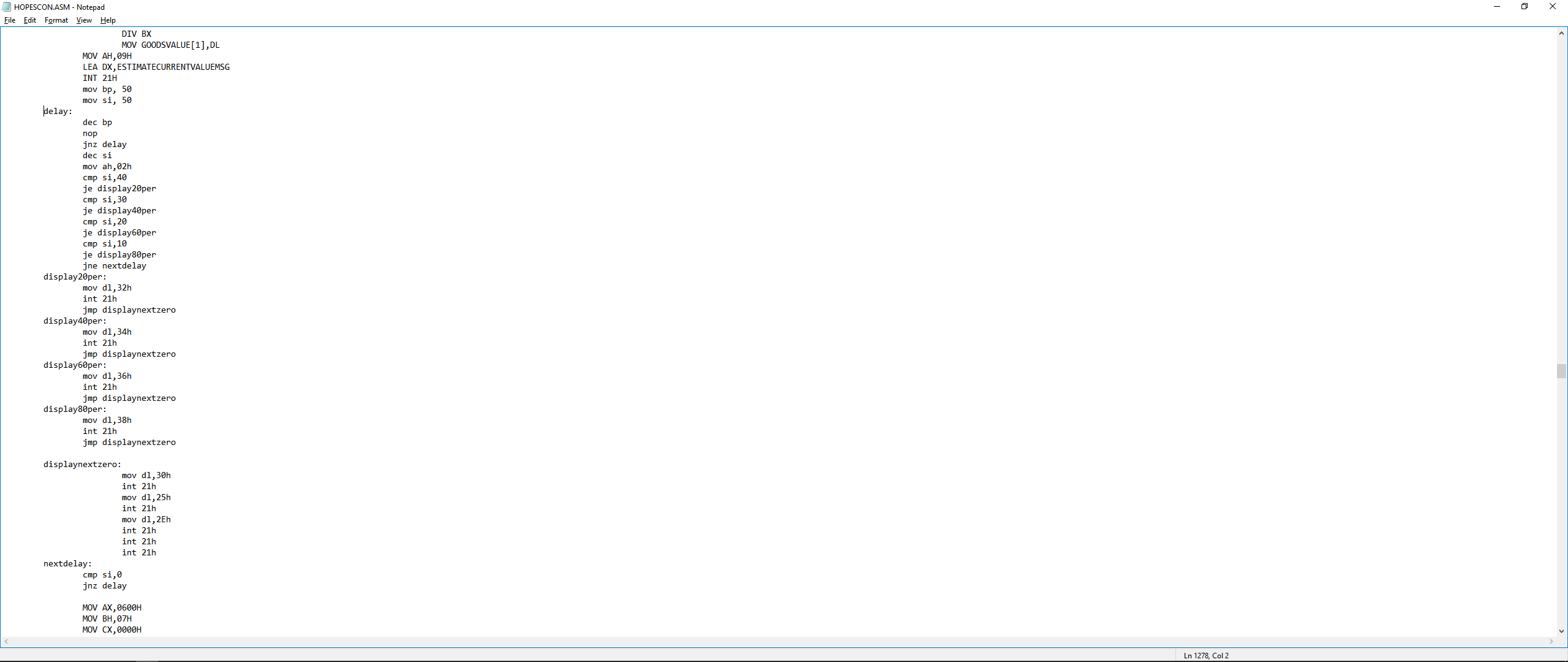
|  |
| --- |
| **AIBIGDATAALGORITHM:**  This function will call 00H function and 1AH interrupt to get the system current time counter since from the last power-on or system reset which will be stored in DX (lower order of clock part). Then the value in DX is moved to AX to be used for division operation. DX is cleared by using XOR operation. Then, 10 is moved to BX then AX is divided by BX. The remainder part (2 bytes value) will be stored in DX, then the value is moved from DL to GOODSVALUE[0] which represent the first digit value of current value of the item. Since is it get from the time counter, so this will be a random value.  **NOGOODSMENU:**  This function is used to determine whether item is in obsolete condition or not. By comparing the random value with 7 using CMP and JB, we can make sure that 7/10 of possibility that item is obsolete so it is easier to demonstrate the process for handling the obsolete item.  **AIBIGDATAALGORITHM2:**  Same with **AIBIGDATAALGORITHM** but run at another moment so will create different random value for the second digit of current value of item. |



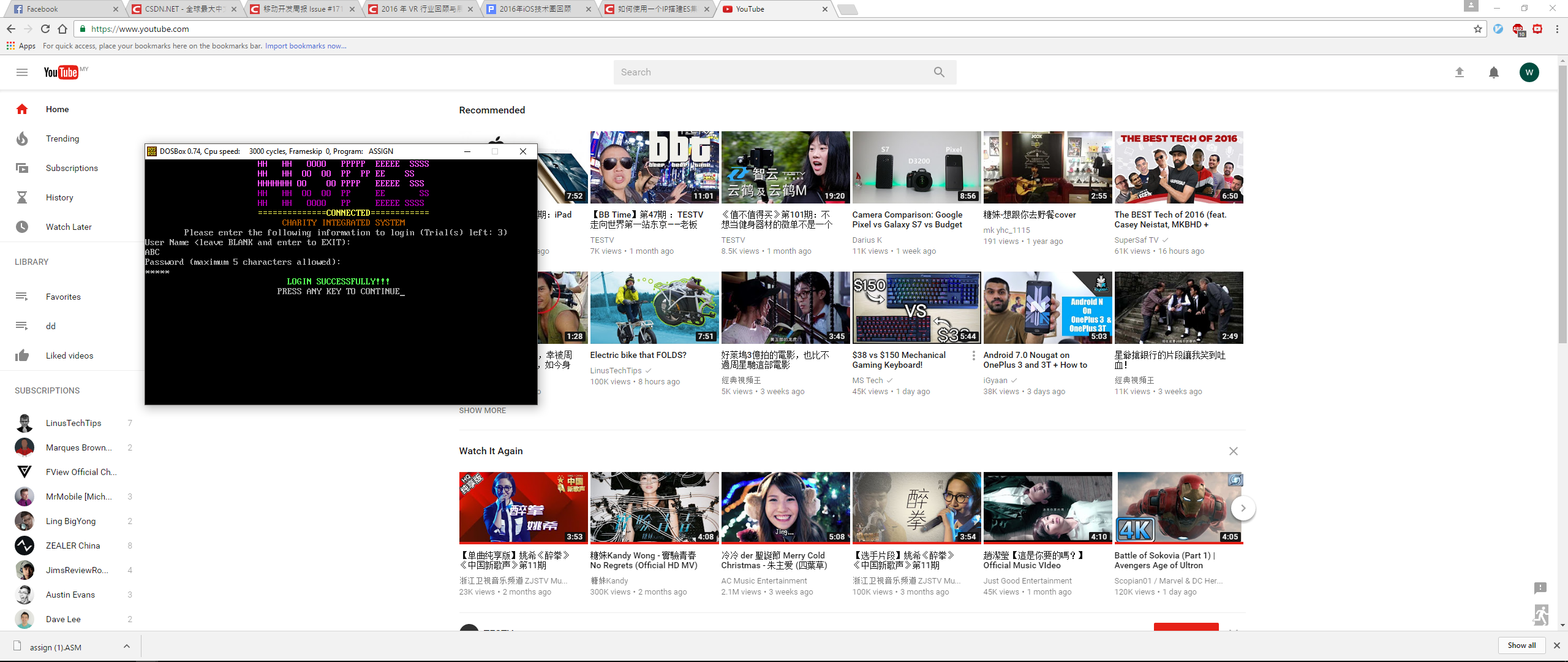




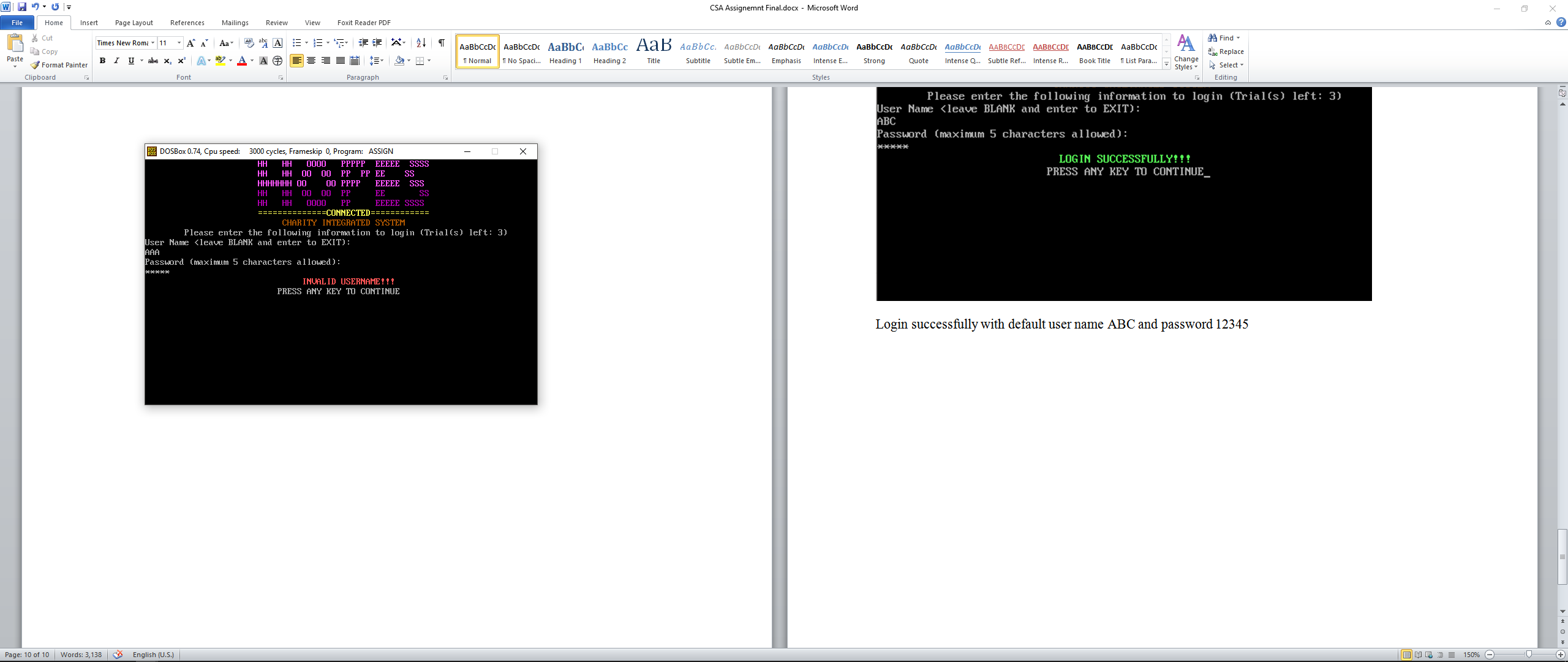
|  |
| --- |
| First three lines are used to display the ESTIMATECURRENTVALUEMSG.    **delay:**  After this, to demonstrate a system estimating effect, a self-defined delay function is used.  First, move 50 to both BP and SI register before the delay label.  Then decrement BP by 1 and then NOP ( No Operation Performed) is used to not change the state of any of the programmer-accessible registers, status flags, or memory and which may require a specific number of clock cycles to execute. This is used for delay timing purpose.  Then decrement SI by 1.  02H function is called to display the 20/40/60/80/nothing depends on the current value of SI.  If SI is not equal to 20 or 40 or 60 or 80, then it will jump to nextdelay  **display20per/ display40per/ display60per/ display80per:**  Display the 2/4/6/8/nothing depends on the current value of SI with int 21h interrupt.  **displaynextzero:**  Display 30h (‘0’), 25h (‘%’) and three 2Eh (‘.’) right after the 2/4/6/8 with int 21h interrupt.  **nextdelay:**  This function is just used to compare SI with 0 to decide when to stop the delay process. |



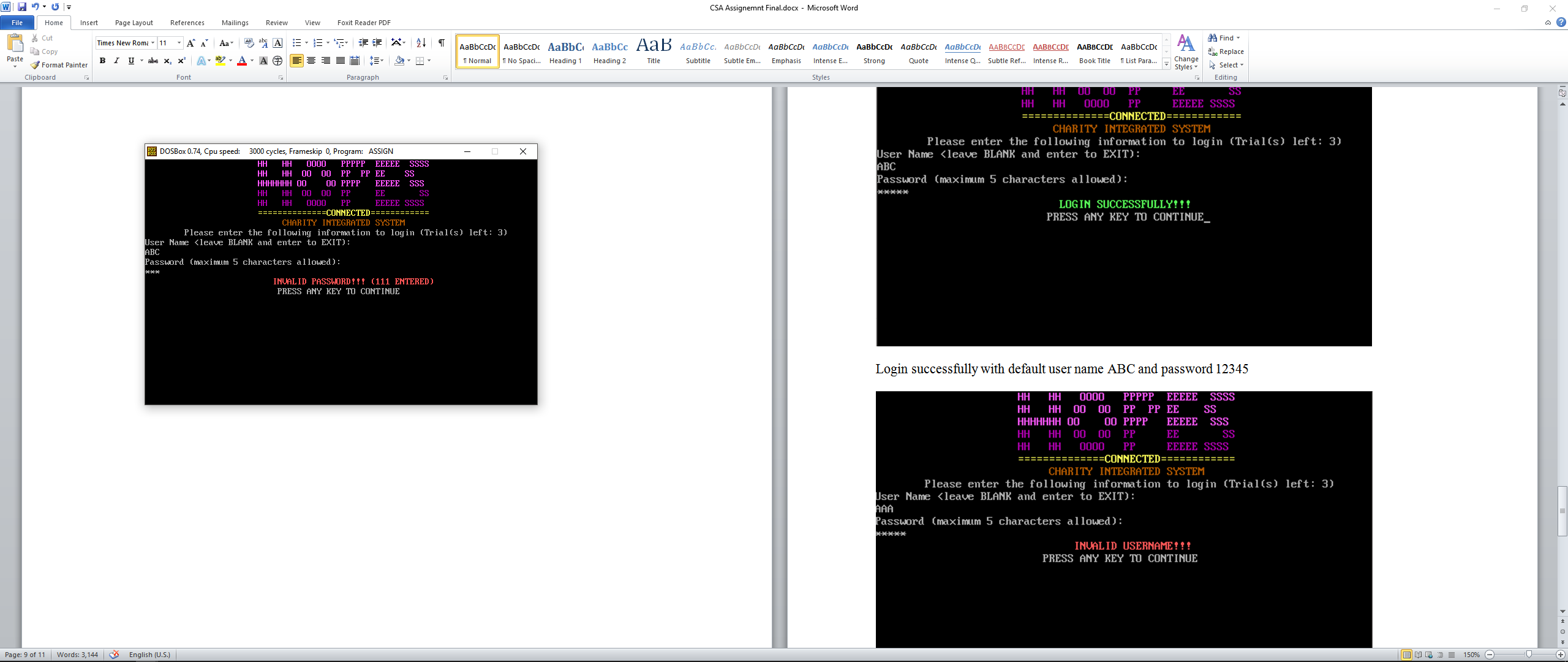
**3. I/O Design**



Login successfully with default user name ABC and password 12345



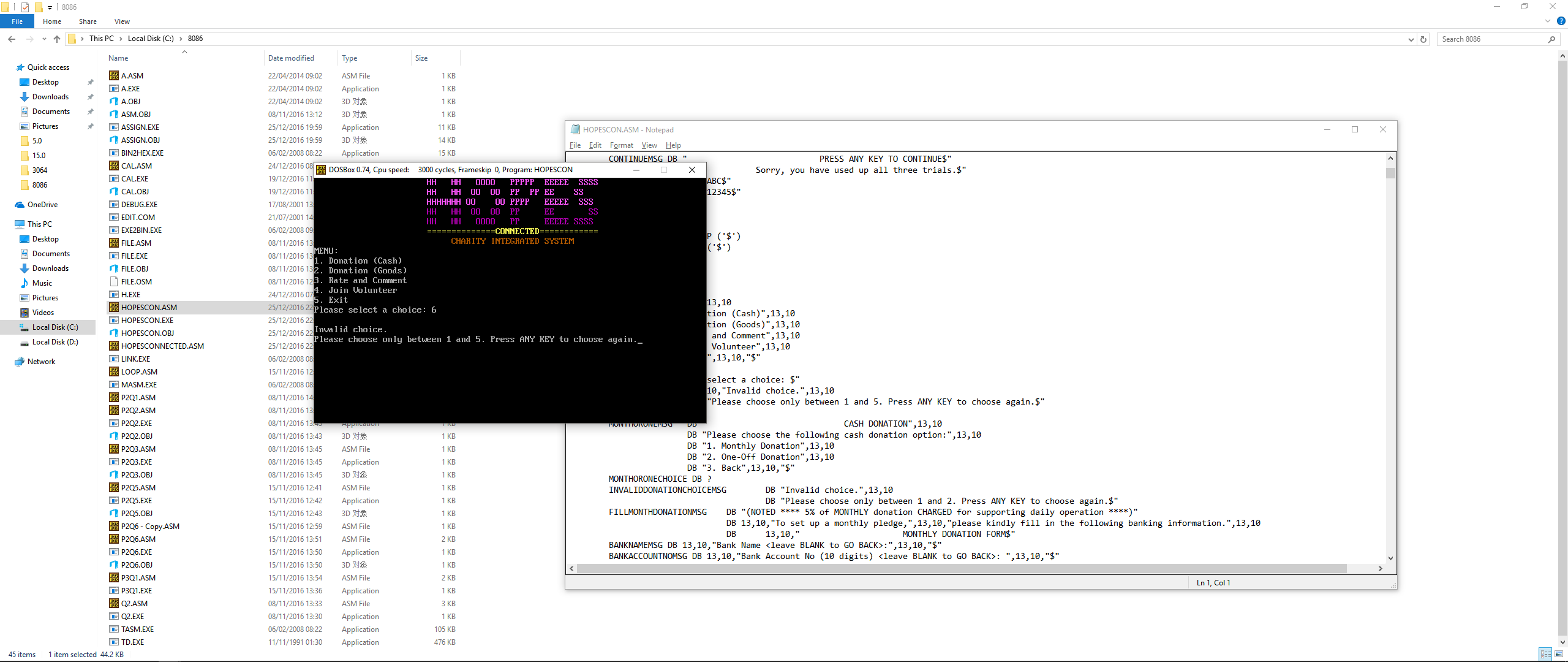
Login unsuccessfully with wrong username AAA



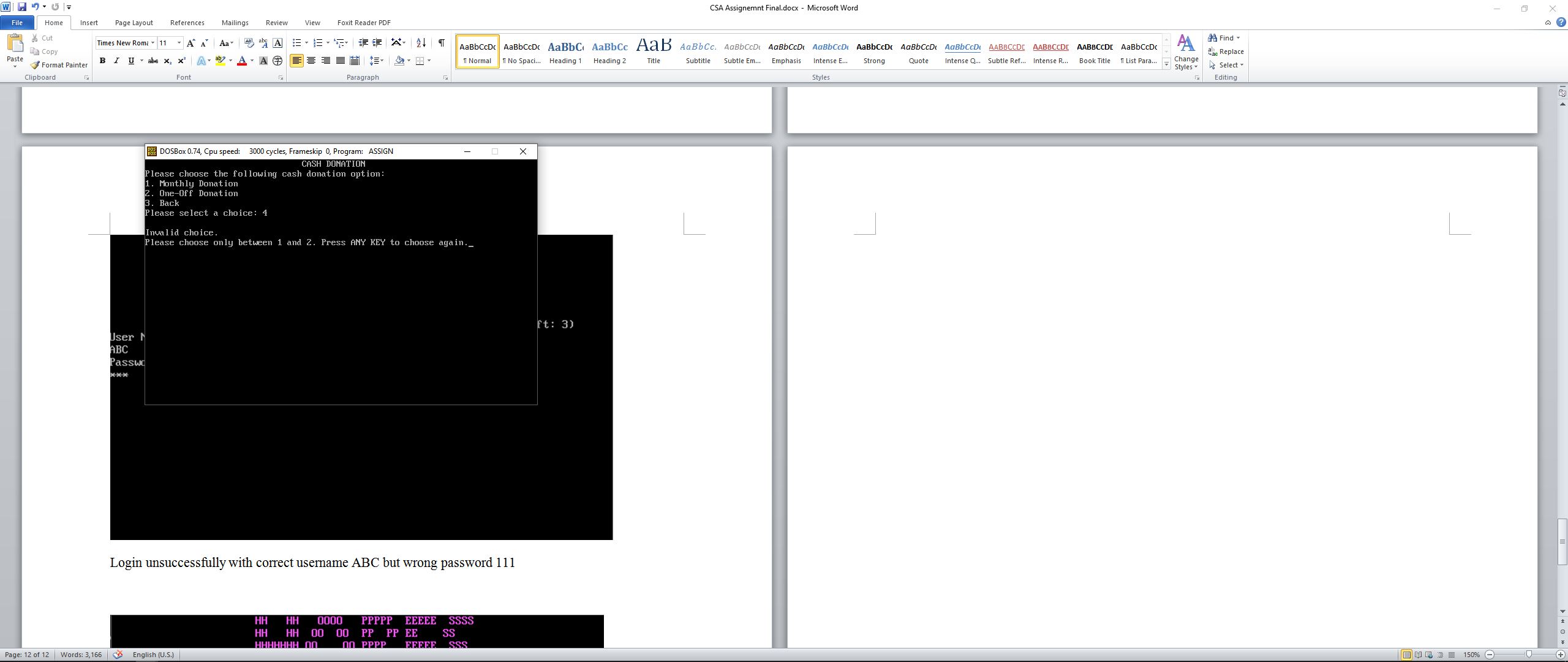
Login unsuccessfully with correct username ABC but wrong password 111



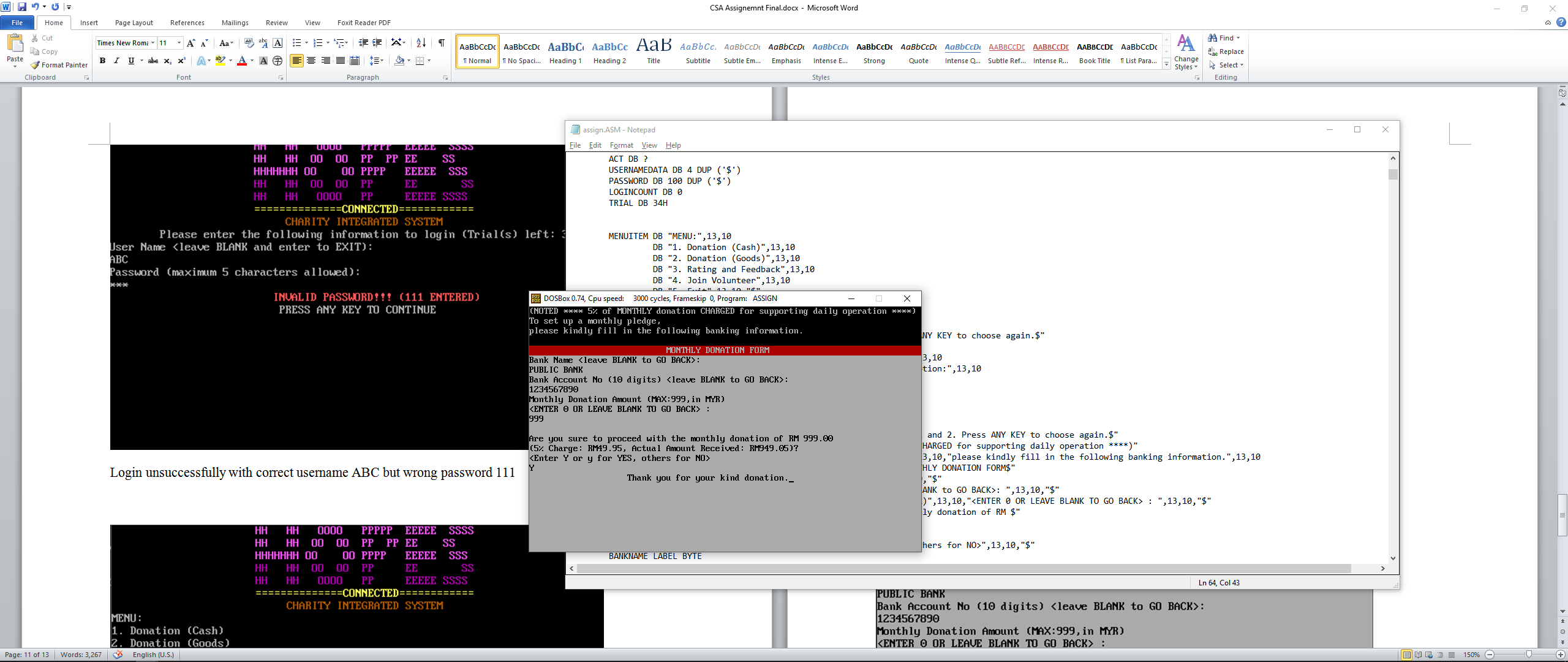
Error message is prompt as user used all up all three trials for login by entering wrong user name or/and password for three times continuously.



Menu for main functions and validation for invalid choice for menu options (valid: 1-5 only)

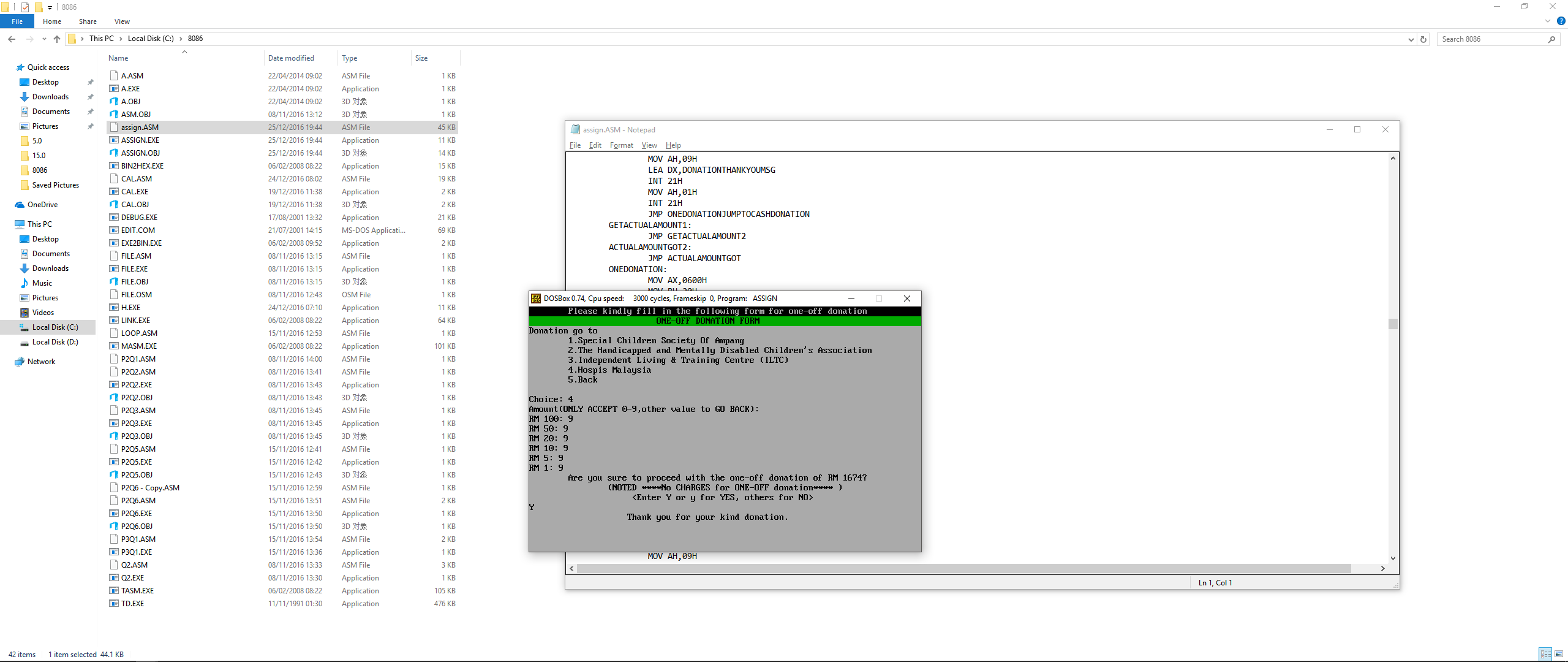


Menu for cash donation and validation for invalid choice for menu options (valid: 1-3 only)



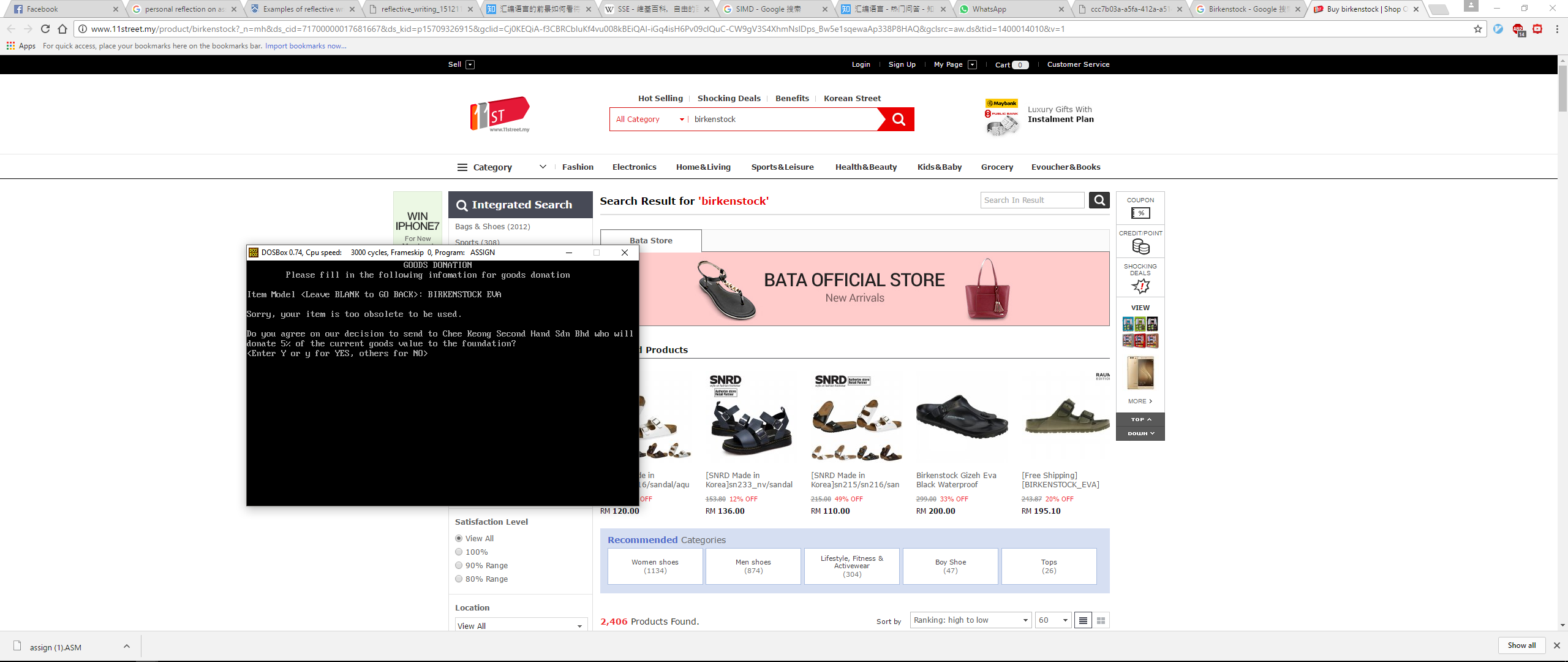
Monthly Donation Form with filled information and confirmation message with calculation results for charges applied to the monthly donation (5% charged to be used for daily operation of foundation)

Thank you message is shown as user entered ‘Y’ to agree with the donation.

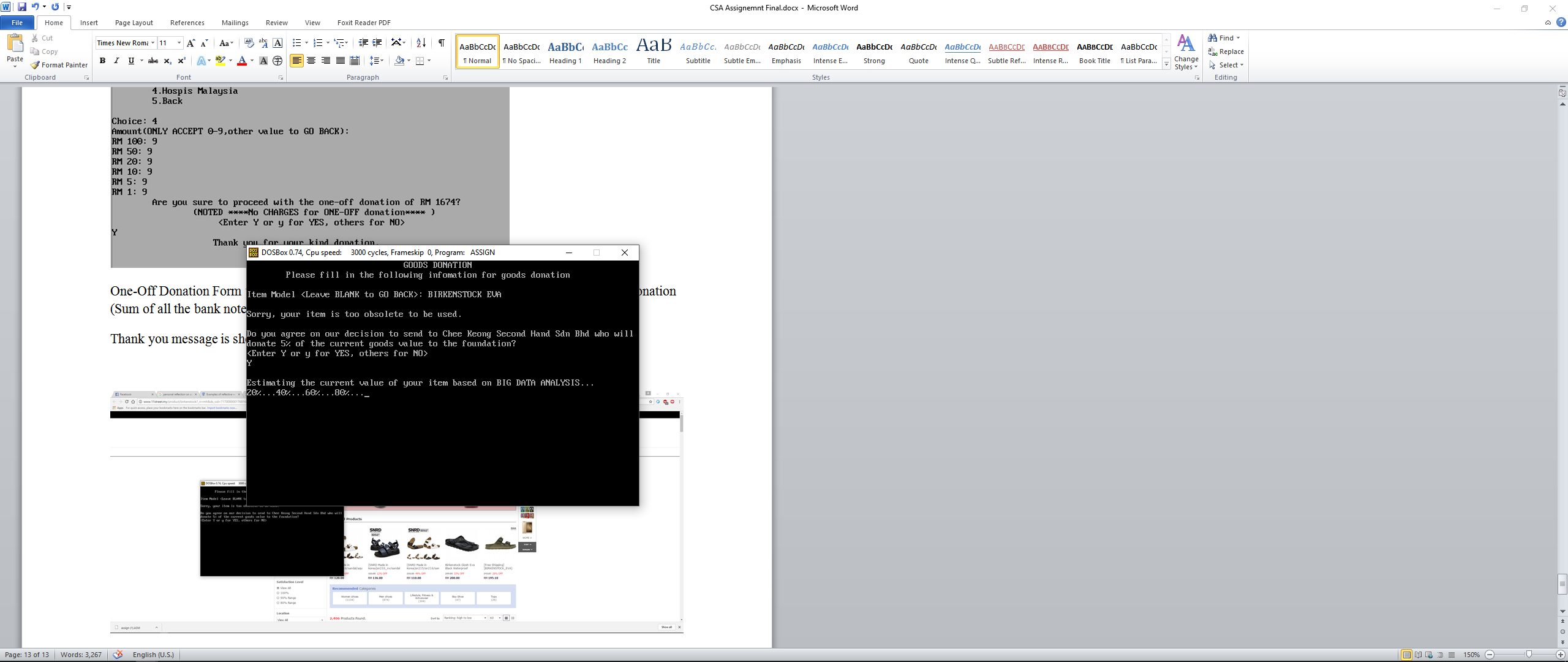


One-Off Donation Form with filled information and confirmation message with total amount of donation (Sum of all the bank notes multiply with their own quantity respectively)

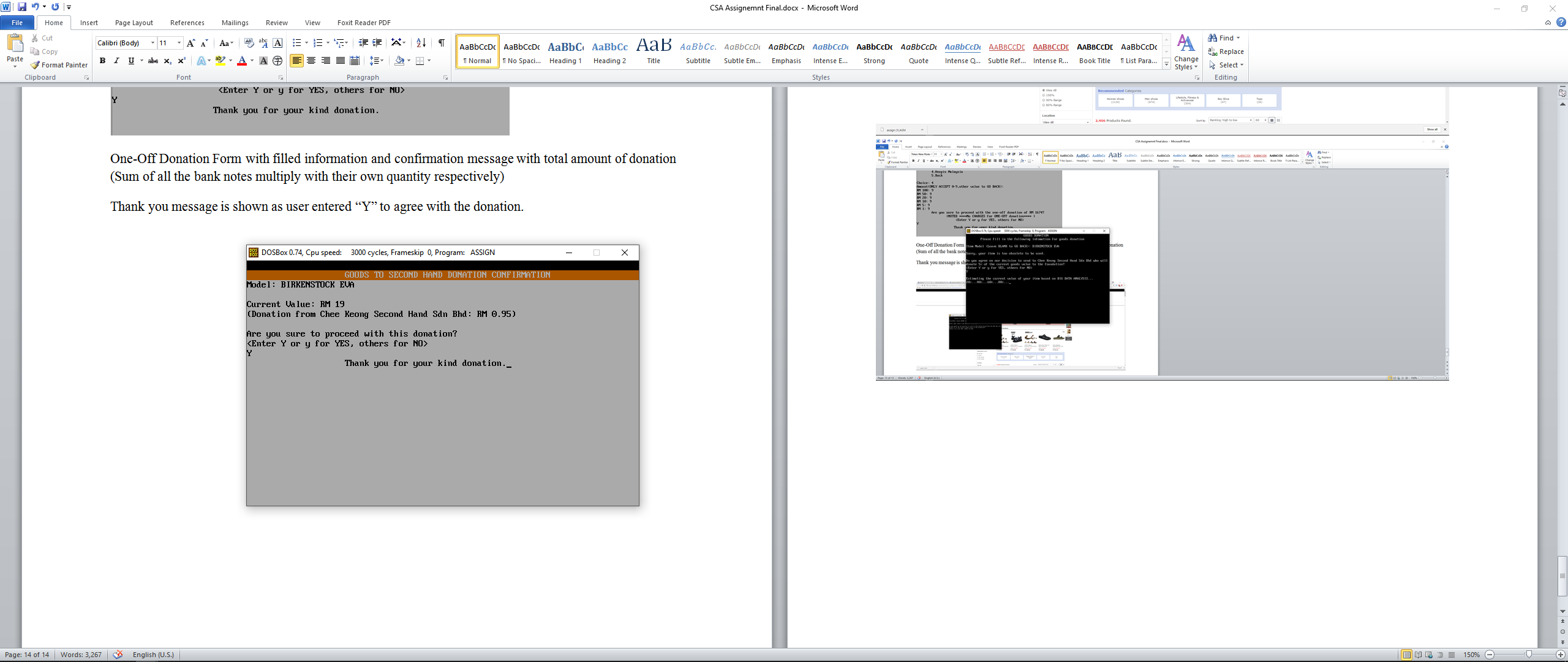
Thank you message is shown as user entered “Y” to agree with the donation.



Goods Donation screen with filled information for item model and system recognize the item as obselete condition. Thus, a confirmation message is displayed to ask user whether they want to sent that item to Chee Keong Second Hand Sdn Bhd who will receive the item and donation 5% of item’s current value to the foundation.

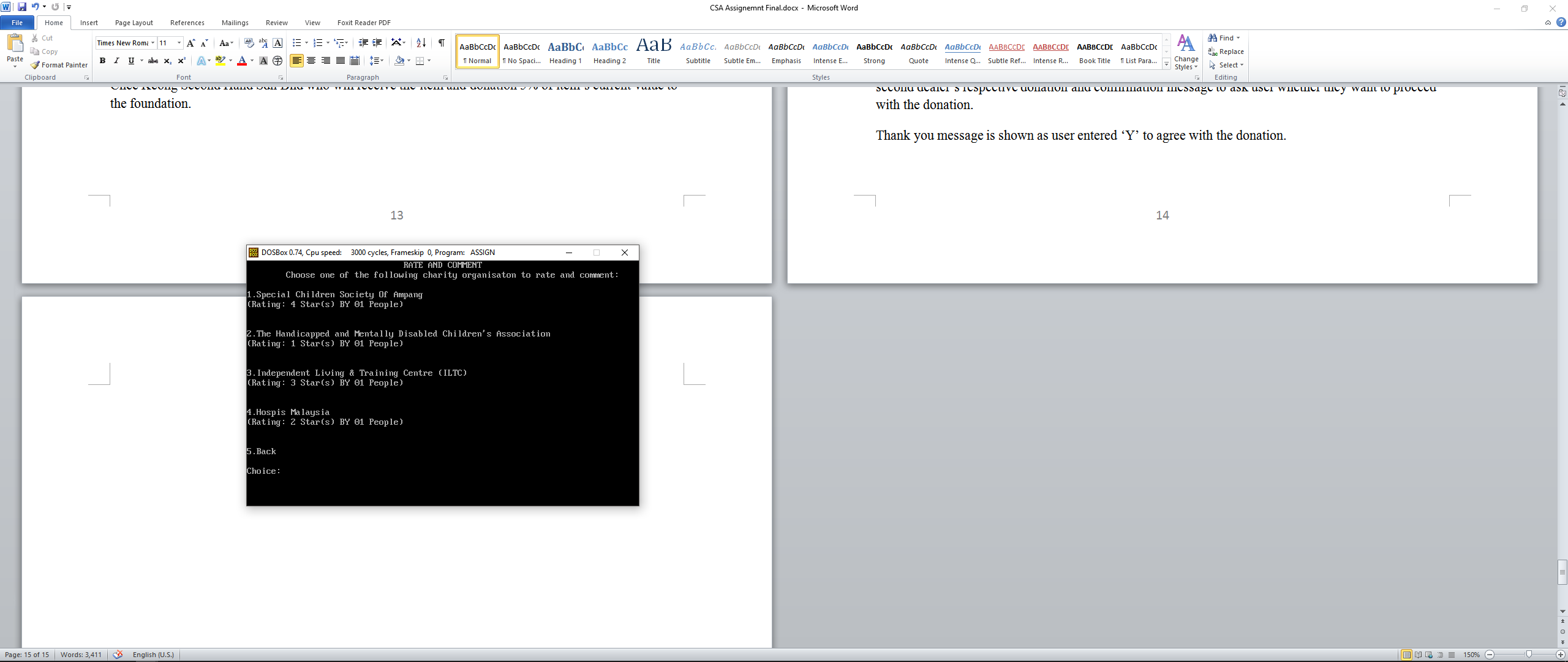


As user entered ‘Y’ for YES, system will estimate the current value of item based on the different market prices from various online stores’ price list and databases using Big Data Analysis which involve the Web crawler and A.I. filtering algorithm.



After that, a Goods To Second Hand Donation Confirmation Form is displayed with item’s current value, second dealer’s respective donation and confirmation message to ask user whether they want to proceed with the donation.

Thank you message is shown as user entered ‘Y’ to agree with the donation.

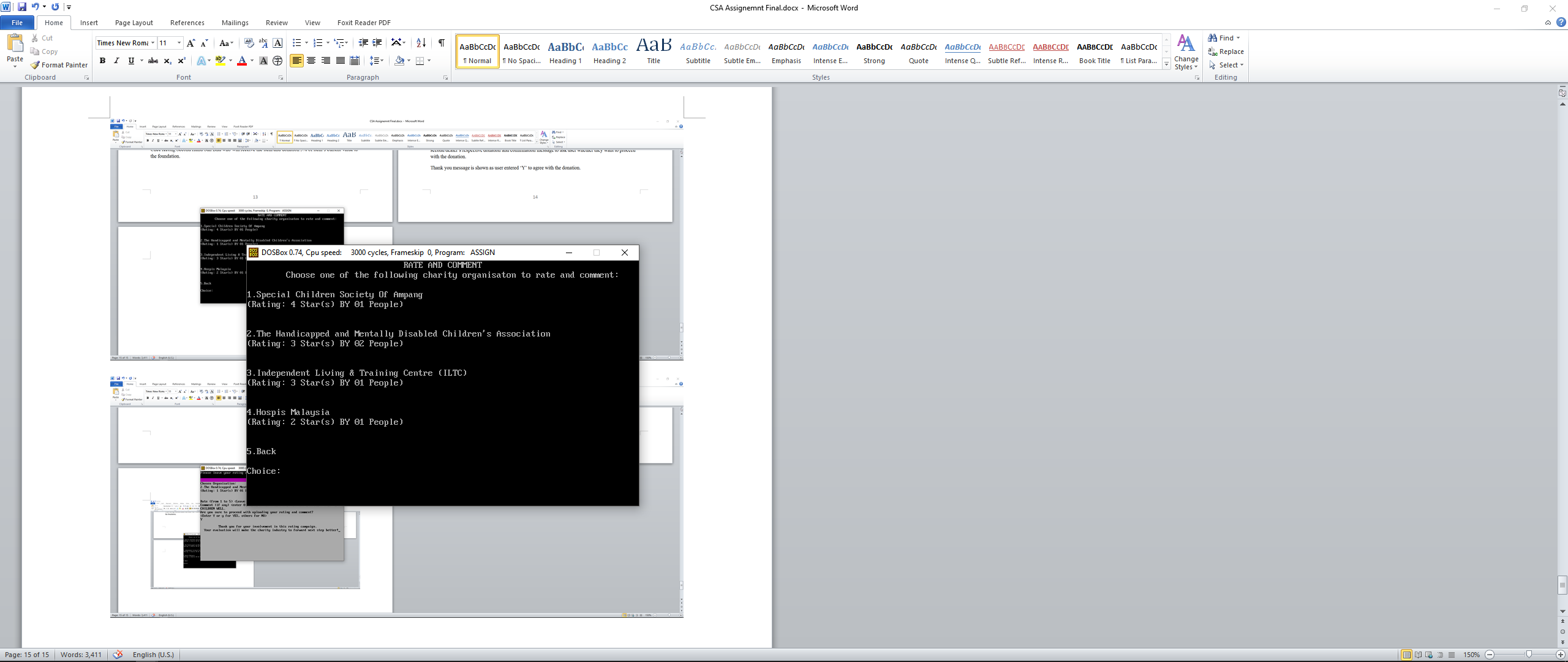


Rate and Comment screen which will display the currently available four organizations to be chosen for rating and leaving comment together with their respective current rating.



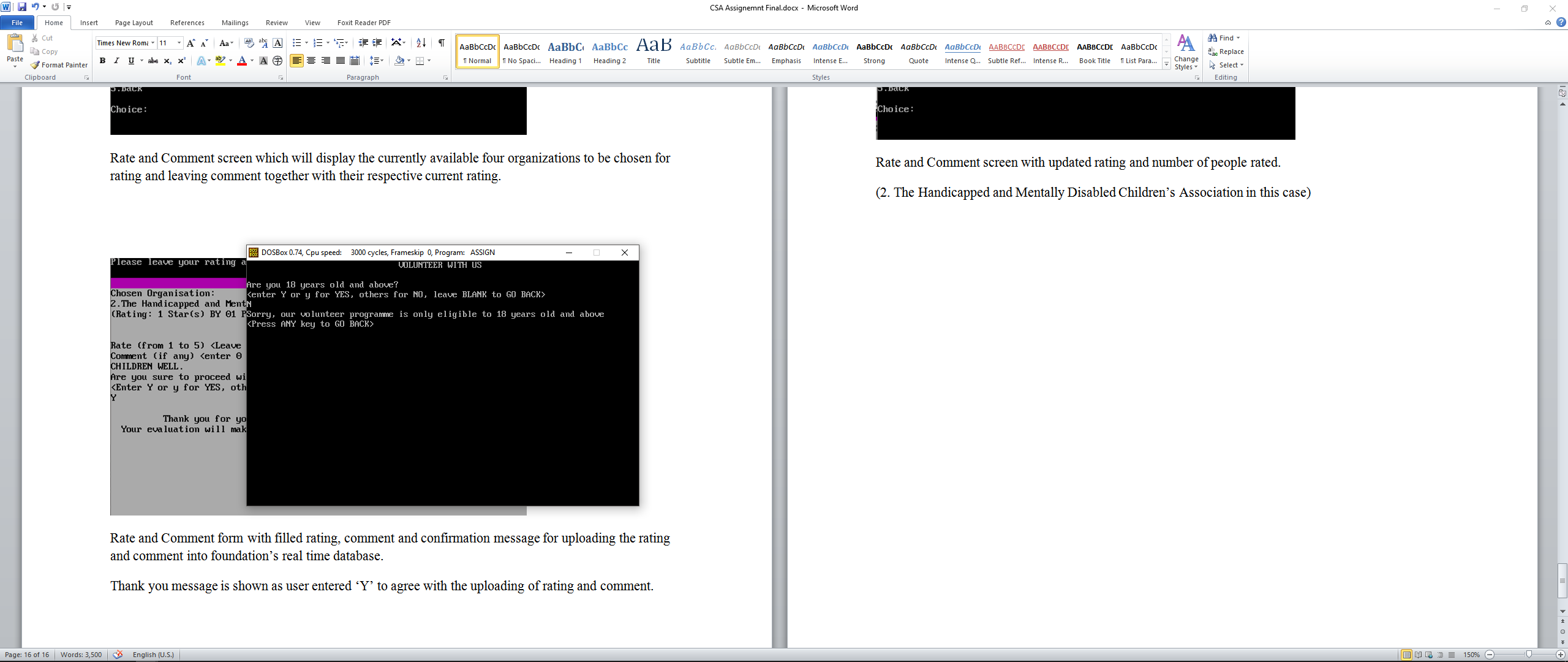
Rate and Comment form with filled rating, comment and confirmation message for uploading the rating and comment into foundation’s real time database.

Thank you message is shown as user entered ‘Y’ to agree with the uploading of rating and comment.



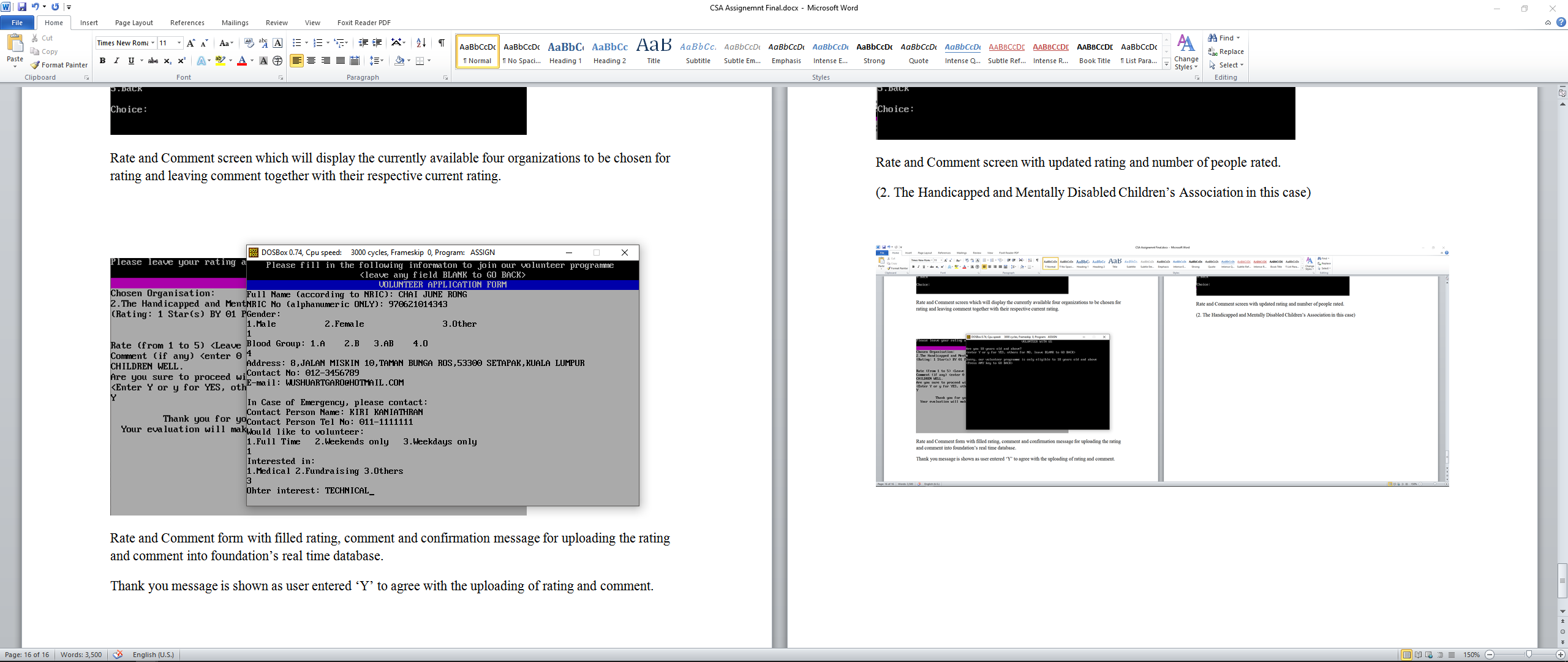
Rate and Comment screen with updated rating and number of people rated.

(2. The Handicapped and Mentally Disabled Children’s Association in this case)

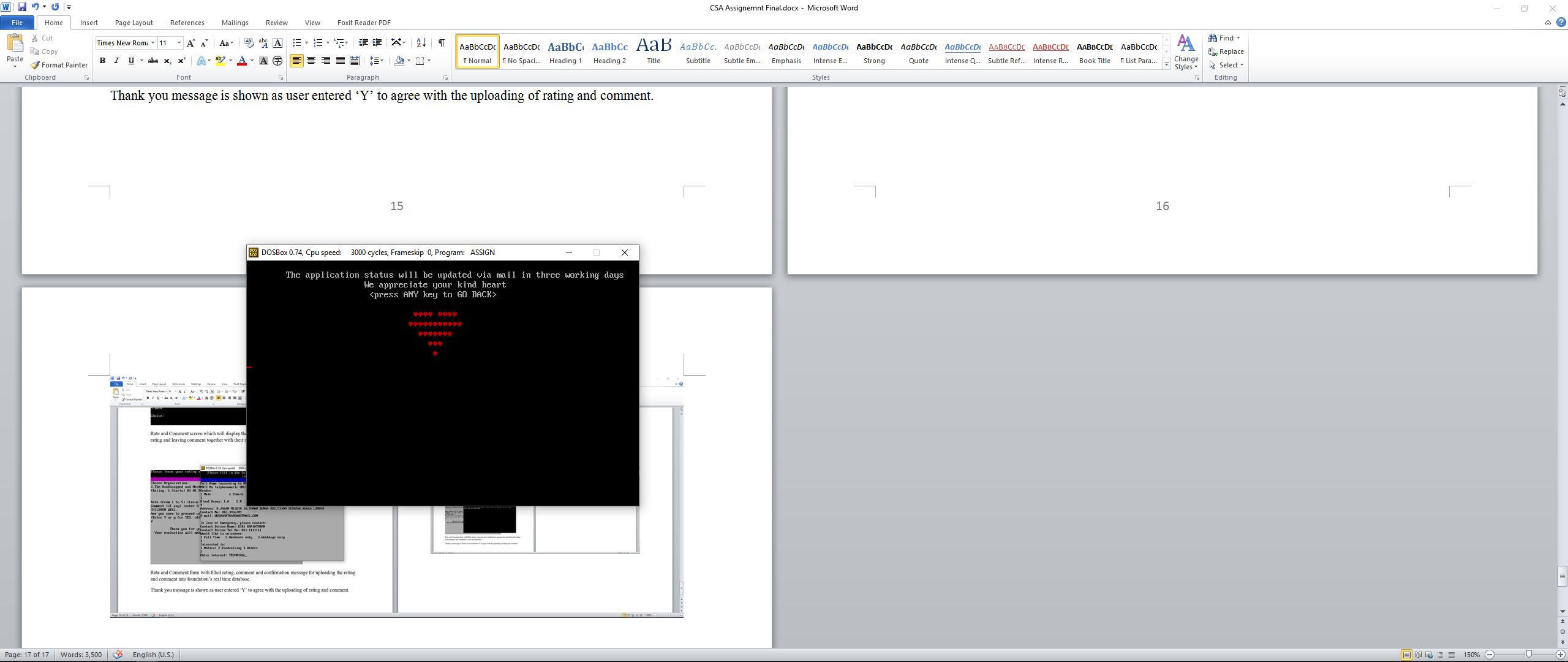


Volunteer with us screen will prompt for the age of the user first before they can proceed to make sure they are 18 years old and above to be eligible as a volunteer for foundation.

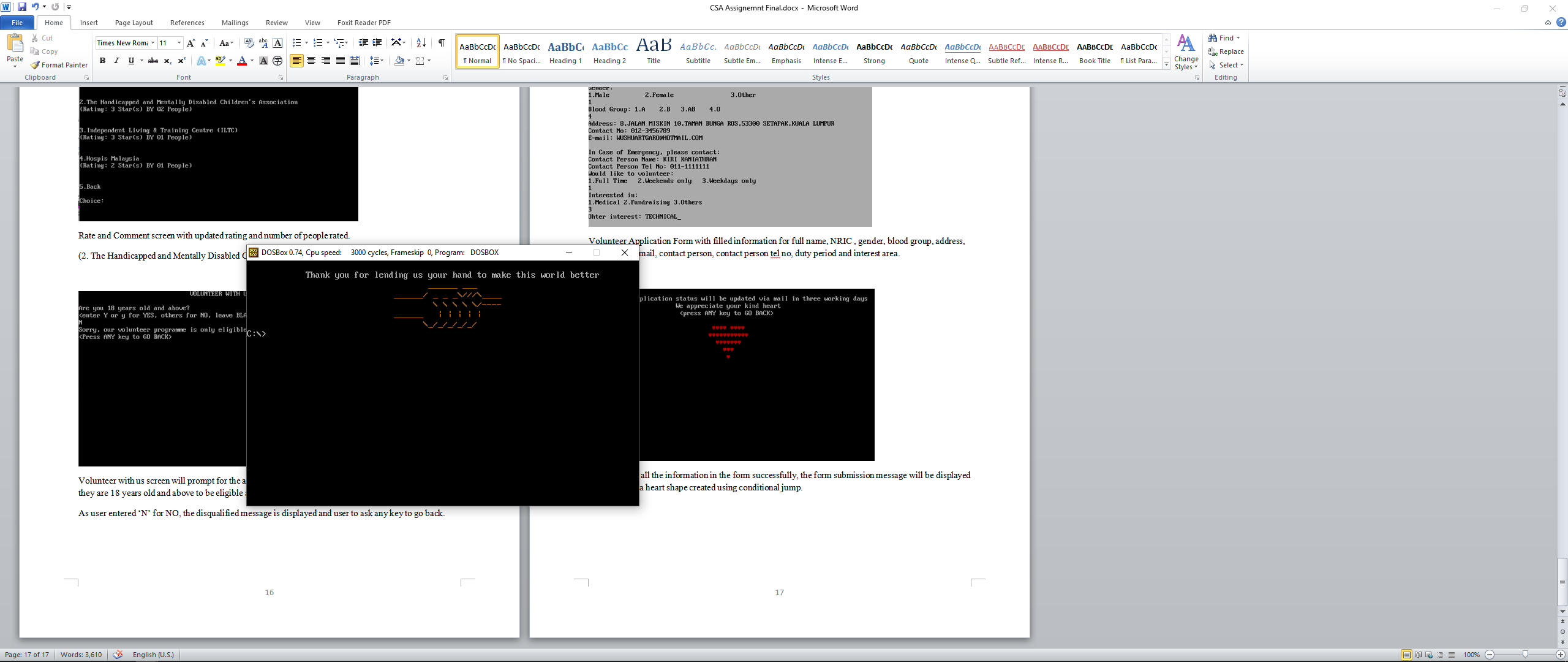
As user entered ‘N’ for NO, the disqualified message is displayed and user to ask any key to go back.



Volunteer Application Form with filled information for full name, NRIC , gender, blood group, address, contact no, e-mail, contact person, contact person tel no, duty period and interest area.



As user filled all the information in the form successfully, the form submission message will be displayed together with a heart shape created using conditional jump.



As per user choose to exit the system from the main menu, a thank you message with hand shake graphic will be displayed.

**4. User Guide**

**Software used:** DOSBox 0.74, notepad, 8086 folder

**Installation / Implementation guide:**

To assemble, link and execute the assembly program and using DOSBox. Please type the following commands in DOSBox (with 8086 folder placed in C:\):

1. MOUNT C C:\8086
2. C:
3. MASM HOPESCON.ASM
4. LINK HOPESCON.OBJ
5. HOPESCON

**Login Information:**

User Name: ABC

Password: 12345

**Step-by-step guideline to use the program**

**LOGIN AND MAIN MENU:**

1. User can enter the ABC as valid user name and 12345 as valid password to log into the system.
2. User name field is limited to 3 characters while password is limited to 5 characters to reduce the careless typing error. Input in the password filed is hidden with asterisk character (\*) for security purpose.
3. Invalid password will be displayed out to notify user for any typing error while valid password won’t be shown for security purpose.
4. There is a trial(s) number left shown on the upper right corner of the screen, it will be decreased every time user failed to login with a wrong name and/or wrong password. After all three trials has been used up, system will exit automatically after showing an error message.
5. If user wants to exit from the login screen, then just leave blank in the user name field and press enter.
6. Choose between 1 to 4 to proceed with different main functions, choose 5 to exit the system
7. Choose 1 for Cash Donation, 2 for Goods Donation, 3 for Rating and Feedback and 4 for Join Volunteer.
8. Other inputs out of 1 to 5 will be invalid and invalid message will be displayed to let user to choose again.

**CASH DONATION:**

1. Choose between 1 to 2 to proceed with different cash donations, choose 3 to go back to the main menu.
2. Choose 1 for Monthly Donation, 2 for One-Off Donation.
3. Other inputs out of 1 to 3 will be invalid and invalid message will be displayed to let user to choose again.

**MONTHLY DONATION:**

1. User can enter the bank name, their respective bank account no. and amount to set up their monthly pledge.
2. The bank account no field is limited to 10 characters while monthly amount field is limited to 3 digits to prevent the value more than maximum amount of RM 999 and system won’t accept any input besides digit to prevent erroneous calculation.
3. For bank name field and bank account no, user can leave blank to go back to cash donation menu. While leaving 0 or blank in the Monthly Donation Amount field will also take user back to cash donation menu.
4. After all the fields is being filled, a confirmation message with calculation results for charges applied to the monthly donation (5% charged to be used for daily operation of foundation) and actual amount of donation being used directly on the unfortunate (Donation – (Donation\*5%)) will be displayed. User can enter ‘Y’ or ‘y’ to proceed with the donation.
5. Thank you message is shown as user entered ‘Y’ to agree with the donation. If user input other character s other than ‘Y’ or ‘y’, system will abandon this donation and jump back to cash donation menu.

**ONE-OFF DONATION:**

1. System will display a list of charity organizations to be chosen by user for this one-off donation to go to. Choose between 1 to 4 to proceed with different charity organizations, choose 5 to exit the system. Other inputs out of 1 to 5 will be invalid and invalid message will be displayed to let user to choose again.
2. Then, system will display different bank notes from RM 100 to RM 1 to let user enter the quantity for each one respectively. System will only accepts input from 0 to 9, other input will take user back to the list of charity organizations. This is to eliminate the erroneous calculation and provide a back option from the bank notes amount input process.
3. After all quantity has been entered, confirmation message with total amount of donation (Sum of all the bank notes multiply with their own quantity respectively) will be displayed. User can enter ‘Y’ or ‘y’ to proceed with the donation.
4. Thank you message is shown as user entered ‘Y’ to agree with the donation. If user input other character s other than ‘Y’ or ‘y’, system will abandon this donation and jump back to cash donation menu.

**GOODS DONATION:**

1. Goods Donation Screen will provide an item model field for user to enter the model of item that they would like to donate. The model can be in terms of code or product name like Birkenstock Eva, C++: From Beginner to Give Up and etc. Then the system will automatically analyze whether that particular item is obsolete to be used anymore or not. (NOTED: the obsolete condition doesn’t depends on the item model but depends on that particular item condition instead). User can leave blank to go back to main menu.
2. If the item is not obsolete then system will display a confirmation message for donation of that item. User can enter ‘Y’ or ‘y’ to proceed with the donation.
3. Thank you message is shown as user entered ‘Y’ to agree with the donation. If user input other character s other than ‘Y’ or ‘y’, system will abandon this donation and jump back to cash donation menu.
4. If the item is obsolete, a confirmation message is displayed to ask user whether they want to sent that item to Chee Keong Second Hand Sdn Bhd who will receive the item and donation 5% of item’s current value to the foundation.
5. As user entered ‘Y’ for YES, system will estimate the current value of item based on the different market prices from various online stores’ price list and databases using Big Data Analysis which involve the Web crawler and A.I. filtering algorithm.The estimating progress will be displayed as

20%...40%...60%...80%...

1. After that, a Goods to Second Hand Donation Confirmation Form is displayed with item’s current value, second dealer’s respective donation and confirmation message to ask user whether they want to proceed with the donation.
2. As user entered ‘Y’ to agree with the donation, a thank you message is shown. If user input other character s other than ‘Y’ or ‘y’, system will abandon this donation and jump back to main menu.

**RATE AND COMMENT:**

1. Rate and Comment screen will display the currently available four organizations to be chosen by user for rating and leaving comment together with their respective current rating.
2. User can choose between 1 to 4 to proceed with different charity organizations, choose 5 to go back to the main menu.
3. Other inputs out of 1 to 5 will be invalid and invalid message will be displayed to let user to choose again.
4. After user choose an organization, then system will display Rate and Comment form which consists of that particular organization’s name and rating status, a rating input field and a comment input field.
5. For rating, user can enter 1 to 5 star(s). Other inputs out of 1 to 5 will be invalid and invalid message will be displayed to let user to enter again. User can leave blank to go back to cash donation menu previous Rate and Comment Screen.
6. For comment field, user can leave it blank or left their own comment for that particular organization.
7. Enter ‘0’ in comment field will take user back to the previous Rate and Comment Screen.
8. After the comment input, a confirmation message for uploading the rating and comment into foundation’s real time database will be displayed. User can enter ‘Y’ or ‘y’ to proceed with the rating and comment update.
9. Thank you message is shown as user entered ‘Y’ to agree with the uploading of rating and comment
10. Rate and Comment screen will be displayed with updated rating and number of people rated.

**JOIN VOLUNTEER:**

1. Volunteer with us screen will prompt for the age of the user first before they can proceed to make sure they are 18 years old and above to be eligible as a volunteer for foundation. As user entered other characters besides ‘Y’ and ‘y’ for NO, the disqualified message is displayed and user to ask any key to go back.
2. As user entered ‘Y’ and ‘y’ for YES, a Volunteer Application Form with fields for full name, NRIC , gender, blood group, address, contact no, e-mail, contact person, contact person tel no, duty period and interest area will be displayed one by one for user to fill in.
3. User can leave blank in any input fields in the form to abandon this application and go back to main menu.
4. As user filled all the information in the form successfully, the form submission message will be displayed together with a heart shape created using conditional jump.

**5. Personal Reflection**

When I decided to really start doing the programming part of this assignment just a week before the deadline, I never thought it would be such a remarkable experience. At first, I was really scared because I would not have enough time to complete it by the deadline as I have never wrote a complete system with assembly language before. However, thing goes the other way round while I really started working on it.

Looking at every code shown on the screen right after my fingers typed on each individual key respectively, I felt an intense satisfaction right in my soul. As the debugging process happened successfully and the inputs entered into the program triggered the desired results, my mind flew into the deepest part of the computer architecture. How impressive it is that we have gone so far, from the Turing Machine born in 80 years ago until A.I. algorithms like AlphaGo that we have today. Learning and using assembly language has made me to have a new perspective on the modern computers. My generation is such a lucky generation that we born in the explosive growth age of personal computer and internet. We never really thought how all those things come from and how do they work internally. The advancement of technology is so remarkable to make us overlook the presence of technology. Every new technology that emerges and every old technology that present just feel as natural as it should be. Meanwhile, we are being greedier and less appreciates towards these great achievements. Take me personally as an example, I never stopped complaining about how inefficient is the Java Virtual Machine’s memory management, how confusing is the pointer concept in C and so on. This assignment had taught me a lesson to re-examine and appreciate the beauty of every programming language.

Programming using assembly language is just similar as construct a building right from the sand and stone, it is inefficient and troublesome in certain extent but it is also a great opportunity and way to allow us to really understand why the modern high level programming languages are needed and why they are being designed like today’s state.

For instance, assembly language allows me to realize that functions with parameters concept in high level programming language is a great way to achieve dynamic and reusable codes. In assembly language, if we would like to perform the same process on different set of data, these data needed to be copied to the temporary variables to go through the same algorithm then assign the result back into their respective data variables. If we don’t do this then we need to repeat the same codes with different data variables which is even more inefficient. By looking at more than 2000 lines of assembly codes to implement a simple workable system and more than 100 lines out of it is just used to implement a simple two decimal places calculation in this assignment, I have deep understanding on the importance of modularity and reusability of a program and why it is being emphasized in most of the high-level programming languages. It is not difficult to imagine that how knotty it would be when we need to modify, maintain or reuse those 2000 lines of codes in another program. Reinventing the wheel may seem to be great for learning the programming right from basic but it will never be a good idea in the real world practice to develop and maintain those complex and enormous software systems.

From my perspective, every programmer should learn the assembly language. It may not be useful from today’s standard but it allows us to really understand what actually is a computer, how to optimize our programs to the hardware level and re-inventing the wheel is a great way to eliminate our boredom so let’s make it our hobby!

* Chai June Rong