**COA Problem SET**

|  |  |
| --- | --- |
|  | Write code fragment to read a character from the keyboard and to display the character on the screen. |
|  | Write a program to a) Prompt the user for Input. b) Read first, middle and last initials of a person’s name. c) Display them down the left margin. **Sample:** **ENTER THREE INITIALS**: **JFK** **J**  **F**  **K** |
|  | Write a program to  a) Display a “?”  b) Read two decimal digits whose sum is less than 10  c) Display them and their sum on the next line, with an appropriate  message.  **Sample:**  **?27**  **THE SUM OF 2 AND 7 IS 9** |
|  | Write a program to read one of the hex digits A-F and display it on  the next line in decimal  **Sample:**  **ENTER A HEX DIGIT: C**  **IN DECIMAL IT IS 12** |
|  | Write a program to display a 10\*10 solid box asterisks.  Hint: Declare a string in the data segment that specifies the box  and display it with INT 21h, function 9h. |
|  | Write an Assembly Program to draw the following pattern using ONLY ONE BYTE TPE VARIABLE. In the following, “\*” is the user input in runtime.   |  |  | | --- | --- | | **Sample input:**  Enter a Character: **\*** | **Sample Output:**  **\***  **\*\*\***  **\*\*\*\*\***  **\*\*\*\*\*\*\*** | |
|  | Write an Assembly Program to draw the following pattern using ONLY ONE BYTE TPE VARIABLE. In the following, “\*” is the user input in runtime.   |  |  | | --- | --- | | **Sample input:**  Enter a Character: **$** | **Sample Output:**  **$**  **$$**  **$$$**  **$$$$**  **$$$$$**  **$$$$**  **$$$**  **$$**  **$** | |
|  | Write a program to display “?”. After that, read four initials and display them as mentioned in the following 6 x 7 box of hashes (#) .  **Sample input**:  Enter Four Initials: **ABCD**  **A#####D**  **#######**  **#######**  **#######**  **#######**  **B#####C** |
|  | Write a program to display “?”. After that, read four initials and display them as mentioned in the following 7 x 7 box of hashes (#) .  **Sample input**:  Enter Four Initials: **ABCD**  **A######**  **#B#####**  **##C####**  **###D###**  **####C##**  **#####B#**  **######A** |
|  | Write a program to display “?”. After that, read four initials and display them as mentioned in the following 7 x 7 box of hashes (#) .  **Sample input**:  Enter Four Initials: **ABCD**  **######A**  **#####B#**  **####C##**  **###D###**  **##C####**  **#B#####**  **A######**  **######A** |
|  | Write a program to display “?”. After that, three four initials and display them as mentioned in the following 5 x 5 box of hashes (#) .  **Sample input**:  Enter Four Initials: **abc**  **#####**  **#####**  **#abc#**  **#####**  **#####** |
|  | Suppose that AX and BX both contain positive numbers and **ADD AX, BX** is executed. Show that there is a carry into the msb but no carry out of the msb if and only if signed over- flow occurs. |
|  | Suppose AX and BX both contain negative numbers and **ADD AX, BX** is executed. Show that there is a carry out of the msb but not into the msb if and only if signed over-flow occurs. |
|  | Write assembly code for each of the following decisions structures.  **IF AX < 0**  **THEN PUT -1 IN BX**  **END\_IF** |
|  | Write assembly code for each of the following decisions structures.  **IF AL < 0**  **THEN PUT FFh IN AH**  **ELSE PUT 0 IN AH**  **END\_IF** |
|  | Write assembly code for each of the following decisions structures.  **(IF DL >= “A”) AND (DL <= ‘2’)\**  **THEN DISPLAY DL**  **END\_IF** |
|  | Write assembly code for each of the following decisions structures.  **IF AX < BX**  **THEN**  **IF BX < CX**  **THEN**  **PUT 0 IN AX**  **ELSE PUT IN BX**  **END\_IF** |
|  | Write assembly code for each of the following decisions structures.  **IF (AX < BX) OR (BX < CX)**  **THEN**  **PUT 0 IN DX**  **ELSE**  **PUT 1 IN DX**  **END\_IF** |
|  | Write assembly code for each of the following decisions structures.  **IF AX < BX**  **THEN**  **PUT 0 IN AX**  **ELSE**  **IF BX < CX**  **THEN**  **PUT 0 IN BX**  **ELSE**  **PUT 0 IN CX**  **END\_IF** |
|  | Find the purpose of the above pseudo-codes. |
|  | Find purpose of the following ASM code   |  |  | | --- | --- | | .MODEL SMALL  .STACK 100H  .DATA  VAR DB 'C'  .CODE  MAIN PROC  MOV AX, @DATA  MOV DS,AX    MOV DL,37H  SUB DL,30H  ADD DL,38H  MOV AH,2  CMP DL,39H  JLE AS    MOV BL,DL | MOV DL,31H  INT 21H  SUB BL,10D  MOV DL,BL  INT 21H    MOV AH,4CH  INT 21H    AS:      INT 21H  MOV AH,4CH  INT 21H  MAIN ENDP  END MAIN | |
|  | Use a CASE structure to code the following:  Read a character,  -If it’s “A” then execute carriage return  -If it’s “B” then execute line feed  -If it’s any other character, then return to DOS |
|  | Write a sequence of instructions to do each of the following:  a) Put the sum 1+4+7+…+148 in AX  b) Put the sum 100+95+90+…+5 in AX |
|  | Write a program to display a “?” ,read four uppercase letters and display them on the next line in alphabetical order. |
|  | Write a program to display a “?” ,read two **lowercase** letters and two **uppercase** letters and display them on the next line in alphabetical order. |
|  | Write a program to display the extended ASCII characters (ASCII codes 80h to FFh). Display 10 characters per line, separated by blanks. Stop after the extended characters have been displayed  once. |
|  | Write a program that will prompt the user to enter a hex digit character(“0” …. ”9” or “A” .… “F”) display it on the next line in decimal and ask the user if he or she wants to do it again. If the  user types “y” or “Y” the program repeats; if the user types anything else the program terminates. If the user enters as illegal character, prompt the user to try again.   |  | | --- | | Enter a hex digit: 9  In decimal is it: 9  Do you want to do it again? Y  Enter a hex digit: c  Illegal character - enter 0-9 or A-F: C  In decimal it is 12  Do you want to do it again? N | |
|  | Write a program that reads a string of capital letters, ending with a carriage return, and displays the longest sequence of consecutive alphabetically increasing capital letters read.   |  | | --- | | Enter a String of Capital Letters: **FABDEKFGHIJC**  The **LONGEST** Consecutively Increasing String Is: **FGHIJ** | |
|  | Write an assembly program to take some input characters from user, sort the alphabets and display them in descending order. |
|  | Write an assembly program to take 4 subjects grade (i.e. A,B, C, B) as input from user and calculate his/her CGPA. Please note, each grade has a difference of 0.5 [Hint: do the average of all subjects]. |
|  | Write an assembly program to take 4 subjects grade (i.e. A,B, C, B) as input from user and calculate CGPA for N students. Please note, each grade has a difference of 0.5 [Hint: do the average of all subjects]. |
|  | Write an assembly program to find whether a character (user input) is VOWEL or CONSONANT or a NUMBER? |
|  | Write an assembly program to find the count VOWEL or CONSONANT or a NUMBER from a given string?  **Sample:**  **ABgC123DdEsFaH**  **3 8 3** |
|  | Write an assembly program that will take 3 separate decimal numbers and store them in three different variables. Now find the biggest number among them. **Sample:** **First number: 6 Second number: 4 Third number: 9 9 is the biggest number.** |
|  | For each of the following instructions, give the new destination contents and the new settings of CF, SF, ZF, PF, and OF. Suppose ·that the flags are initially 0 in each part al this question.  a. ADD AX, BX where AX contains 7FFFh and BX contains 0001h  b. SUB AL, BL where AL contains 01h and BL contains FFh  c. DEC AL where AL contains 00h  d. NEG AL where AL contains 7Fh  e. XCHG AX,BX where AX contains 1ABCh and BX contains 712Ah  f. ADD AL,BL where AL contains 80h and BL contains FFh  g. SUB AX,BX where AX contains 0000h and BX contains 8000h  h. NEG 'AX where AX contains 0001h |
|  | Suppose ADD AX, BX is executed. In each of the following parts, the first number being added is the contents of AX, and the second number is the contents of BX. Give the resulting value of AX and tells whether signed or unsigned overflow occurred.  a) 512Ch + 4185h  b) FE12h + 1ACBh  c) E1E4h + DAB3h  d) 7132h + 7000h  e) 6389h + 1176h |
|  | Suppose SUB AX, BX is executed. In each of the following parts, the i1rst number i5 the initial contents of AX and the second number is the contents of BX. Give the resulting value of AX and tell whether signed or unsigned overflow occurred.  a) 2143h – 1986h  b) 81FEh – 1986h  c) 19BCh – 81FEh  d) 0002h – FE0Fh  e) 8BCDh – 71ABh |
|  | Write an assembly program that will only exit when the user will press three consecutive same inputs otherwise it will just take input and show the symbol as output. Before the end of the program it will give a message ‘**Thank you. Program ends here**.’  **Sample:**  **Please enter anything: 6**  **You have entered: 6**  **Please enter anything: 6**  **You have entered: 6**  **Please enter anything: 6**  **You have entered: 6**  **Thank you. Program ends here.** |
|  | Write a program that prompts the user to type a binary number of 16 digits or less, and outputs it in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again. |
|  | Write a program that prompts the user to enter two binary numbers of up to 8 digits each and prints their sum on the next line in binary. If the user enters an illegal character, he or she should be prompted to begin again. Each input ends with a carriage return. |
|  | Write a program that prompt the user to type a hex number of four hex digits or less and outputs it in binary on the next line. If the user enters an illegal character, he or she should be prompted to begin again. Accept only uppercase letters. |
|  | Write a program that prompts the user to enter two unsigned hex numbers, 0 to FFFh and prints their sum in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again. Your program should be able to handle the possibility of unsigned overflow. Each input ends with a carriage return. |
|  | Write a program that prompts the user to enter a character, and on subsequent lines prints its ASCII code in binary, and the number of 1 bit in its ASCII code.  **Sample:**  **TYPE A CHARACTER: A**  **THE ASCII CODE OF A IN BINARY IS: 01000001**  **THE NUMBER OF l BITS IS: 2** |
|  | Write a program that prompts the user to type a hex number of four hex digits or less, and outputs it in binary on the next line. If the user enters an illegal character, he or she should be · prompted to begin again. Accept only uppercase letters. Also, your program may ignore any Input beyond four characters.  **Sample:**  **TYPE A HEX NUMBER (0 TO FFFF): 1a**  **ILLEGAL HEX DIGIT, TRY AGAIN: lABC**  **IN BINARY IT IS 0001 1010 1011 1100** |
|  | Write a program that prompts the user to enter a string of decimal digits, ending with a carriage return, and prints their sum in hex on the next line. If the user enters an illegal character, he or she should be prompted to begin again.  **Sample:**  **ENTER A DECIMAL DIGIT STRING: 1299843**  **ENTER A DECIMAL DIGIT STRING: 1299843** |
|  | Suppose that AX= 1234h, BX= 5678h, CX = 9ABCh, and SP= 100h.Give the contents of AX, BX, CX, and SP after executing the following instructions:  PUSH AX  PUSH BX  XCHG AX, CX  POP CX  PUSH AX  POP BX  When the stack has completely filled the stack area, SP = 0. If another word is pushed onto the stack, what would happen to SP? What might happen with the stack after execution each instruction the program? |
|  | Write some code to  a. place the top of the stack into AX, without changing the stack contents.  b. place the word that is below the stack top into CX. without changing the stack contents. You may use AX.  c. exchange the top two words on the stack. You may use AX and BX |
|  | Suppose a program contains the lines  CALL PROC1  MOV AX, BX and  Instruction MOV AX, BX is stored at 08FD:0203h  PROCI Is a NEAR procedure that begins at 08FD:300h,  SP= 010Ah.  What are the contents of SP just after CALL PROCl is executed? What word is on top of the stack? |
|  | Write a program that lets the user type some text, consisting of words separated by blanks, ending with a carriage return and display the text in the same word order as entered but with the letters in each word reversed. For example : “This is a test” becomes “siht si a tset”. |
|  | Write a program that displays a “?” calls READ to read a binary number and calls RANDOM and WRITE to compute and display 100 random numbers. The numbers should be displayed four per line, with four blanks separating the numbers. |
|  | Write a procedure to find the maximum and minimum number from set of numbers |
|  | Write a procedure that will find the count of vowel, consonant, number from given string of characters |
|  | Write a program that lets the user type in an algebraic expression ending with a carriage return that contains round (Parenthesis), square and curly brackets. As the expression is being typed in the program evaluates each character. If at any point the expression is incorrectly bracketed (too many right brackets or a mismatch between left and right brackets), the program tells the user to start over. After the carriage return is typed, if the expression is correct, the program displays”EXPRESSION IS CORRECT”. If not the program displays “TOO MANY LEFT BRACKETS”. In both cases the program asks the user if he or she wants to continue. If the user types “Y”, the program runs again. The program does not need to store the input string, only check it for correctness. |
|  | Write a program that lets the user to enter time in seconds up to 65535 and outputs the time as hours, minutes and seconds. Use INDEC and OUTDEC to do the I/O. |
|  | Write a program to find the greatest common divisor (GCD) of two integers M & N, according to the following algorithm:  a) Divide M by N, getting quotient Q and remainder R.  b) If R = 0, stop. N is the GCD of M & N.  c) If R <> 0, replace M by N, N by R, and repeat step 1.  Use INDEC to enter M and OUTDEC to print GCD. |
|  | Write a program to make calculator that can perform addition, subtraction, multiplication, division. [Procedure for each operation] |
|  | Write a program that uses XLAT to  a) Read a line of text,  b) Print it on the next line with all small letters converted to capital letters,  punctuation, digit characters and so on. |
|  | Write a procedure PRINTHEX that uses XLAT to display the content of BX as four-digit hex integer, stores it in BX and calls PRINTHEX to print it on the next line. |