**AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH**

**List of Experiments and Projects**

Faculty of Science and IT

Name of the Lab: Computer Organization and Architecture (LAB)

Course Title: Computer Organization and Architecture

|  |  |
| --- | --- |
| SN | Title Of the experiment/Projects |
| **Write programs using Assembly Language for the following experiments** | |
| 1 | We will identify the x86 arithmetic instructions and examine a program which will  perform the following arithmetic operations:  Calculate : (30 10 + 15 10 ) \* ( 575 10 – 225 10 ) + 210  Each time after pressing the “single step” button, check and record down the contents of AX  and BX registers in Table 1 |
| 2 | In this example, we will develop a program to perform the following arithmetic operations: 1210  \*( 20010 –22510 )+127  Each time after single-stepping, observe and record down the contents of AX and BX  registers in Table 2. |
| 3 | Write an Assembly Program to Swap Two numbers Entered by User: |
| 4 | Write an Assembly Program to Add Two Integers Entered by User. Each input should be greater than or equals to 5. |
| 5 | Write an Assembly Program to print the following message:  “The Sum of 3 and 2 is 5 (using ADD). However, if we multiply 3 and 2 the product will be  6 (using MUL). On the other hand if we subtract 3 and 3 , the result will be 1(using SUB)”  In the sentence above the all the numbers will be user input and you should print the whole  line utilizing the value of those variables. |
| 6 | Write an Assembly Program to print the following pattern: |
| 7 | Write a program to display “?” (Question mark), read two decimal digits whose sum is less than 10 and display the digits and their sum on the next line with an appropriate message.  **Sample execution:**  ? 25  The sum of 2 and 5 is 7 |
| 8 | Write a program to prompt the user to put the initials of name, read first middle and last initials and display them down the left margin  **Sample execution:**  Enter Three Initials: JFK  J  F  K |
| 9 | Write a program to read one of the hex digits A-F, and display it on next line in decimal.  **Sample execution:**  Enter a HEX digit: A  The Decimal value of A is: 10 |
| 10 | Write a program to display a 10 x 10 rsolid box of hashes (#). [Hint: declare a string using array and display it using int 21h function 9h].  **Sample Execution:**  ##########  ##########  ##########  ##########  ##########  ##########  ##########  ##########  ########## |
| 11 | Write a program to display “?” read three initials and display them in the middle of an 7 x 7 box of hashes (#) and beep the computer.  **Sample Execution:**  Enter Three Initials: JFK  #######  #######  #######  ##JFK##  #######  #######  ####### |
| 12 | Write a program to display a "?", read two capital letters, and display them on the next line in alphabetical order. |
| 13 | Write a program to display the extended ASCII characters (ASCJI codes 80h to FFh). Display 10 characters per line, separated by blanks. Stop after the extended characters have been displayed once. |
| 14 | 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 an illegal character, prompt the user to try again.   **Sample Execution:**  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 |
| 15 | Do the programming-exercise 14(above), except that if the user fails to enter a hex-digit character In three tries, display a message and terminate the program. |
| 16 | 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.  **Sample Execution:**  Enter a String of Capital Letters: **FGHADEFGHC**  The **LONGEST** Consecutively Increasing String Is: **DEFGH** |
| 17 | Write an assembly program to take some input characters from user, sort the alphabets and display them in descending order. |
| 18 | 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]. |
| 19 | Do the above problem-18 for N number of students. |
| 20 | Write an assembly program to find whether a character (user input) is VOWEL or CONSONANT or a NUMBER? |
| 21 | Write an assembly program to create a simple calculator that will facilitate the features like Addition, Subtraction, Multiplication and Division. If you decide the operation based on a number input, instruct your software users to avoid mistakes (i.e. prompt messages). Also, if a user gives a wrong input, show the ERROR message and ask for input again. |
| 22 | 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 execution:**  TYPE A CHARACTER: **A**  THE ASCII CODE OF A IN BINARY IS: **01000001**  THE NUMBER OF l BITS IS: **2** |
| 23 | Write a program that prompts the user to enter a character and prints the ASCII code of the character in hex on the next line. Repeat this process until the user types a carriage return.  **Sample execution:**  TYPE A CHARACTER: Z  THE ASCII CODE OF Z IN HEX IS: 5A  TYPE A CHARACTER: |
| 24 | 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 execution:**  TYPE A HEX NUMBER (0 TO FFFF): **1a**  ILLEGAL HEX DIGIT, TRY AGAIN: **1ABC**  IN BINARY IT IS **0001 1010 1011 1100** |
| 25 | 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. Also, your program may ignore any input beyond 16 characters.  **Sample execution:**  TYPE A BINARY NUMBER, UP TO 16 DIGITS: **11100001**  IN HEX IT IS **E1** |
| 26 | 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.  **Sample execution:**  TYPE First BINARY NUMBER, UP TO 8 DIGITS: **11001010**  TYPE Second BINARY NUMBER, UP TO 8 DIGITS: **10011100**  THE BINARY SUM IS: **101100110** |
| 27 | Write a. program that prompts the user to enter two unsigned hex numbers, 0 to FFFFh, 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.  **Sample execution:**  TYPE A HEX NUMBER, 0 - FFEF: 21AB  TYPE A HEX NUMBER, 0 - FFFF: FE03  THE·SUM IS: 11FAE |
| 28 | 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 execution:**  ENTER A DECIMAL DIGIT STRING: 1299843  THE SUM OF THE DIGITS IN HEX IS: 0024 |
| 29 | Write an assembly program to take 4 input characters from user, sort the alphabets and display them in descending order. |
| 30 | Write an assembly program to create a simple calculator that will facilitate the features like Addition, Subtraction, Multiplication and Division. If you decide the operation based on a number input, instruct your software users to avoid mistakes (i.e. prompt messages). Also, if a user gives a wrong input, show the ERROR message and ask for input again. |
| 31 | 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:  **Sample Input:**  TYPE A BINARY NUMBER, UP TO 16 DIGITS: 11110011  **Sample Output:**  IN HEX IT IS: F3 |
| 32 | Write a. program that prompts the user to enter two unsigned hex numbers, 0 to FFFFh, and prints their sum in hex on the next line. Each input ends with a carriage return.  **Sample Input:**  TYPE A HEX NUMBER, 0 - FFEF: 189A  TYPE A HEX NUMBER, 0 - FFFF: B102  **Sample Output:**  THE SUM IS: C99C |
| 33 | Write 2 procedures for a reversed application. The INPUT procedure will take a string input from user (until a CR is found) and the REVERSE procedure will print the reversed output in next line.  **Sample Input:**  Please enter a String: ABCD  **Sample Output:**  The reversed value of the input is = CDBA  In addition, write a third procedure named FACTORIAL that prints the factorial value of a user input.  **Sample Input:**  Please enter a value: 5!  **Sample Output:**  The factorial of 5 is = 120(This is an ASCII code) |
| 34 | Stack Tracing   |  |  |  | | --- | --- | --- | | .MODEL SMALL |  |  | | .STACK 100H |  |  | | .DATA |  |  | | .CODE |  |  | | MYPROC PROC |  |  | | PUSH BP | **[SP] = ( )H** |  | | MOV BP, SP | **[BP]= ( )H** |  | | MOV AH,2 | **[SP+2] = ( )H** |  | | MOV DX, [BP+4] | **[SP] = ( )H** |  | | INT 21H |  |  | | MOV DX, [BP+6] | **BP = ( )H** | **[BP+4] = ( )H** | | INT 21H |  |  | | POP BP | **SP = ( )H** | **[SP] = ( )H** | | RET |  |  | | MYPROC ENDP |  |  | | MAIN PROC |  |  | | MOV AX,61H |  |  | | PUSH AX |  |  | | MOV AX,62H |  |  | | PUSH AX |  |  | | CALL MYPROC | **SP = ( )H** | **[SP] = ( )H** | | MOV AH,4CH |  |  | | INT 21H |  |  | | MAIN ENDP |  |  | | END MAIN |  |  | |