

CS 3443 Computer systems

Assignment 03: Full marks 100

Due date:10/02/2022 (11:59 PM CT)

End date: 10/09/2022 (11:59 PM CT)

Please consider these points below you proceed with the assignment:

- Follow the instructions provided on each question.
- Make sure to follow how the grade is distribution for each question. It is given in green text inside the square bracket.
- For assembly programming, just making the program run is not important. You need to follow the right concept.
- You cannot write your own MIPS program that satisfy the program description and/or output. Your MIPS program should be equivalent to the given C program, following the instructions given below the program.

Following problem is focused on conditional statement in MIPS.

1. Write a complete running equivalent MIPS programs for the following C program: [19 Marks]

```
/*
Program description: Check whether a character is a vowel or a consonant
*/

1. #include <stdio.h>
2. int main() {
3.     char c;
4.     int lowercase_vowel = 0, uppercase_vowel = 0; // [0.5 mark]

5.     printf("Enter an alphabet: "); // [0.25 mark]
6.     scanf("%c", &c); // [0.5 mark]

7.     // check if variable c is a lowercase vowel
8.     if (c == 97 || c == 101 || c == 105 || c == 111 || c == 117) // [5.5 mark]
9.         lowercase_vowel = 1; // [0.25 mark]

10.    // check if variable c is a uppercase vowel
11.    if (c == 65 || c == 69 || c == 73 || c == 79 || c == 85) // [5.5 mark]
12.        uppercase_vowel = 1; // [0.25 mark]

13.    // evaluates to 1 (true) if c is a vowel
14.    if (lowercase_vowel || uppercase_vowel) // [0.5 mark]
15.        printf("%c is a vowel.", c); // [0.25 mark]
16.    else
17.        printf("%c is a consonant.", c); // [0.25 mark]

18.    return 0; // [0.25 mark]
19. }
```

Instructions:

- The value of the variable "c" should be stored in the stack, after reading the character from the user at line # 6. [2.5 mark]
- When comparing the value of the variable "c" at line # 8 and line # 11, its value should be brought from the stack to the register. [2.5 mark]

Following problems are focused on simple loop in MIPS.

2. Write a complete running equivalent MIPS programs for the following C program: [13 Marks]

```
/*
Program description: Print ASCII characters between the given range
*/

1. #include <stdio.h>
2. int main() {
3.     char lowerRange, upperRange, c;

4.     printf("Please enter the first character: ");           // [0.25 mark]
5.     scanf(" %c", &lowerRange);                             // [0.25 mark]

6.     printf("Please enter the last character: ");           // [0.25 mark]
7.     scanf(" %c", &upperRange);                             // [0.25 mark]

8.     printf("The alphabets between %c and %c are: ", lowerRange, upperRange); // [0.25 mark]
9.     for (c = lowerRange + 1; c < upperRange; ++c)         // [2.25 mark]
10.        printf("%c ", c);                                   // [0.25 mark]

11.    return 0;                                              // [0.25 mark]
12. }
```

Instructions:

- The variables, *lowerRange* and *upperRange*, should be stored in the user data segment part of the MIPS memory. [2.5 + 2.5 mark]
- Each value of the variable, *c*, should be stored in the stack and updated there. [4 mark]

3. Write a complete running equivalent MIPS programs for the following C program: [13 Marks]

```
/*
Program description: Find GCD of two whole numbers
*/

1. #include <stdio.h>
2. int main(){
3.     int n1, n2, i;
4.     int gcd;

5.     printf("Enter two whole numbers greater than 0: ");   // [0.25 mark]
6.     scanf("%d %d", &n1, &n2);                             // [0.5 mark]

7.     for(i = 1; i <= n1 && i <= n2; ++i){                  // [3.5 mark]
8.         // ensure if i is a factor of both integers
9.         if(n1 % i == 0 && n2 % i == 0)                     // [2.5 mark]
10.            gcd = i;                                       // [0.5 mark]
11.    }

12.    printf("G.C.D of %d and %d is %d", n1, n2, gcd);       // [0.25 mark]

13.    return 0;                                              // [0.25 mark]
14. }
```

Instructions:

- The value of the variable "gcd" should be stored in the stack at line # 10. [2.75 mark]
- To print the value of gcd in line # 25, its value has to be brought from the stack. [2.5 mark]

Following problems are focused on arrays and pointers in MIPS.

4. Write a complete running equivalent MIPS programs for the following C program: [9 Marks]

```
/*
Program description: Access Array Elements Using array index
*/

1. #include <stdio.h>
2. int main() {
3.     int data[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}; // [1 mark]

4.     printf("The array elements are: \n"); // [0.25 mark]
5.     for (int i = 0; i < 10; ++i) // [2 marks]
6.         printf("%d\n", data[i]); // [2.5 mark]

7.     return 0; // [0.25 mark]
8. }
```

Instruction:

- The integer array data in line # 3 should be stored in the stack. [3 mark]
- To get the full point on line # 6, you need to use array index to access the array elements.

5. Write a complete running equivalent MIPS programs for the following C program: [9 Marks]

```
/*
Program description: Access Array Elements Using Pointers
*/

1. #include <stdio.h>
2. int main() {
3.     short data[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}; //[1 mark]

4.     printf("The array elements are: \n"); // [0.25 mark]
5.     for (int i = 0; i < 10; ++i) // [2 mark]
6.         printf("%hi\n", *(data+i)); // [2.5 mark]

7.     return 0; // [0.25 mark]
8. }
```

Instruction:

- The short integer array data in line # 3 should be stored in the stack. [3 mark]
- To get the full point on line # 6, you need to use pointer to access the array elements.

6. Write a complete running equivalent MIPS programs for the following C program: **[8 Marks]**

```
/*
Program description: Find the length of a String and the array starting address
*/

1. #include <stdio.h>
2. int main()
3. {
4.     // Char array of size 100 can hold a string of 100 chars
5.     char str[100]; // [1 mark]
6.     char i;

7.     printf("Please enter a string: \n"); // [0.25 mark]
8.     fgets(str, 100, stdin); // [0.5 mark]

9.     // '\0' represents the end of String in C
10.    for(i=0; str[i]!='\0'; ++i); // [2.25 mark]
11.    printf("\nLength of input string: %d",i); // [0.25 mark]

12.    printf("\nThe array starts at the memory location: %p", &str); // [0.5 mark]

13.    return 0; // [0.25 mark]
14. }
```

Instruction:

- The character array str in line # 5 should be stored in the stack. **[3 mark]**

Following problem is focused on nested loops in MIPS.

7. Write a complete running equivalent MIPS programs for the following C program: **[29 Marks]**

```

/*
Program description: Program to display a pattern
*/

1. #include <stdio.h>

2. int main(void) {
3.     int n;

4.     printf("Please enter the number of columns:"); // [0.25 mark]
5.     scanf("%d",&n); // [0.25 mark]

6.     //printing the upper part of the pattern.
7.     for(int i=0;i<n;i++){ // [4 mark]
8.         for(int j=0;j<i;j++){ // [4 mark]
9.             printf(" "); // [0.25 mark]
10.        }
11.        for(int k=1;k<=n-i;k++){ // [4 mark]
12.            printf("%d", k); // [0.5 mark]
13.        }
14.        printf("\n"); // [0.25 mark]
15.    }

16.    //printing the lower part of the pattern.
17.    for(int i=1;i<n;i++){ // [4 mark]
18.        for(int j=1;j<n-i;j++){ // [4 mark]
19.            printf(" "); // [0.25 mark]
20.        }
21.        for(int k=1;k<=i+1;k++){ // [4 mark]
22.            printf("%d", (i+2 - k)); // [0.5 mark]
23.        }
24.        printf("\n"); // [0.25 mark]
25.    }

26.    return 0; // [0.25 mark]
27. }

```

Instruction:

- The variable 'n' in line # 3 needs to be stored in the user data segment part of the MIPS memory. [2.25 mark]

Submission Guidelines:

1. All home assignments will be submitted **ONLY** through Canvas.
2. The program should be submitted in .asm file.
 - a. Each of your .asm file need to have header information, i.e. it should include: full name, official email address, date and brief description of the program
 - b. Make sure you use comments and appropriate label/variables name. Codes without comments will be given partial credit. Up to 50% of the marks would be deducted for not writing sufficient comments.
3. When naming your home assignments, please use the naming convention below:
Assignment03_firstName_lastName_Question01.asm
4. All the codes should also be copied in textual form and save into pdf format. Please note that the code should not be copied in image form or saved as an image file.
Assignment03_firstName_lastName_Question01.pdf

Failure to submit in this order will automatically results in 10 points deduction.