IE 332 - Homework #1

Due: Sept 12th, 11:59pm EST

Read Carefully. Important!

As outlined in the course syllabus this homework contributes to your project grade. The maximum attainable mark on this homework is **80**. As was also outlined in the syllabus, there is a **zero tolerance** policy for any form of academic misconduct. Each group submits one assignment.

By electronically submitting this assignment all members of the group acknowledge these statements and accept any repercussions if in any violation of ANY Purdue Academic Misconduct policies. You must upload your homework on time for it to be graded. No late assignments will be accepted. **Only the last uploaded version of your assignment will be graded**.

NOTE: You should aim to submit no later than 30 minutes before the deadline, as there could be last minute network traffic that would cause your assignment to be late, resulting in a grade of zero.

You must use the provided LaTeX template on Brightspace to create the pdf your assignment. No exceptions, at penalty of assignment grade of zero.

1. This question is meant to reinforce your understanding of how a CPU executes program code by focusing on the low-level instructions it executes, versus higher-level instructions you would use in a programming language such as R. You will use the hypothetical assembly language outlined below in Table 1, in addition to 10 registers referred to by $\$r0, \ldots, \$r15$, and system RAM as needed.

instruction	**meaning**	**example**
add reg1, reg2, reg3	$reg1 = reg2 + reg3$	add $r0, $r1, $r2
sub reg1, reg2, reg3	$reg1 = reg2 - reg3$	sub $r0, $r1, $r2
div reg1, reg2, reg3	$reg1 = reg2/reg3$	div $r0, $r1, $r2
divi reg1, reg2, reg3	$reg1 = reg2/reg3$, truncates fractional values	divi $r0, $r1, $r2
mul reg1, reg2, reg3	$reg1 = reg2 * reg3$	mul $r0, $r1, $r2
muli reg1, reg2, x	$reg1 = reg2 * x$, where x is some integer	mul $r0, $r1, 4
addi reg1, reg2, x	$reg1 = reg2 + x$, where x is some integer	addi $r0, $r1, 5
subi reg1, reg2, x	$reg1 = reg2 - x$, where x is some integer	subi $r0, $r1, 5
divr reg1, reg2, x	$reg1 = reg2/x$, where x is real, truncates result	divr $r0, $r1, 2
mod reg1, reg2, reg3	$reg1 = reg2$ mod $reg3$	mod $r0, $r1, $r2
mov reg1, reg2	$reg1 = reg2$	mov $r0, $r1
label:	create a reference label in the code	main:
name: .type, value	create *.type* variable in RAM where $name = value$	var1: .word 5
lw reg, RAM_source	move word from RAM to a register	lw $r4, var1
sw reg, RAM_source	move word from register to RAM	sw $r4, var1
la reg, RAM_source	move address of RAM to a register	la $r4, var1
sa reg, RAM_source	move address in register to RAM	sa $r4,var1
b label	jump to a label in the program	b main
beq reg1,reg2,label	jump to label if $reg1 = reg2$	beq $r0,$r1,main
blt reg1,reg2,label	jump to label if $reg1 < reg2$	blt $r0,$r1,main
bgt reg1,reg2,label	jump to label if $reg1 > reg2$	bgt $r0,$r1,main
bqe reg1,reg2,label	jump to label if $reg1 \geq reg2$	bqe $r0,$r1,main
ble reg1,reg2,label	jump to label if $reg1 \leq reg2$	ble $r0,$r1,main
bne reg1,reg2,label	jump to label if $reg1 \neq reg2$	bne $r0,$r1,main
print reg	print contents of register to screen	print $r0
prints reg	print string pointed to in register to screen	prints $r0
read reg	get keyboard input (after ENTER pressed); store in reg	read $r0
.wordspace size	allocates mory of size integers, all initialized to zero	arr: .wordspace 10
done	end of program	

Table 1: Hypothetical assembly language

*NOTE: for `lw` if `RAM_source` is an integer instead of a memory address, or reg1 and/or reg2 in `beq`, `blt`, `bgt`, `bge`,`ble`, `bne`, then it will be automatically interpreted as such. For instance, `lw $r1, 4`, will be an equivalent instruction to setting $r1=4. Also, potential variable types include `.word` (for integers or floats), `.bit` (for bits), `.asciiz` (for strings/character arrays - noting that each character has an associated bit pattern and can be interpreted as an integer as well). Furthermore, a sequence of comma separated integers is accepted as valid input, allowing you to create integer arrays, e.g., `arr: .asciiz 10,13,12`. Arrays are indexed from 1, and accessing a value from an array is done by referring to the `RAM_source` as `array_name[index]` (e.g., if loading an array value, use `lw $r1, A[1]` where A is the array name). The number of elements in an array is stored at position 0, (e.g. A[0]).

Ex. computing the first n Fibonacci numbers ($F_n = F_{n-1} + F_{n-2}$ for $n > 0$, $F_1 = 1$ and $F_2 = 1$) can be done with the following code (NOTE: this was created from scratch, not translated from another language):

```
string1:  .asciiz "The result is: "
value:    .word 0
main:     read $r9
          ble $r9, 0, end
          lw $r0, value
          addi $r0, $r0, 1
          beq $r9, 1, end
          mov $r2, $r0
          mov $r1, $r2
          lw $r8, 1
loop:     add $r0, $r1, $r2
          mov $r2, $r1
          mov $r1, $r0
          addi $r8, $r8, 1
          beq $r9, $r8, end
          b loop
end:      sw $r0, value
          la $r7, string1
          prints $r7
          print $r0
          done
```

(a) (20 points) Translate the following C++ program into assembly instructions without adding personal insight or design modifications - translate it in a way that a compiler would. When submitting your results show a table with two columns: (1) the original C program, (2) your assembly code. You can increase the page margins or reduce font size to 10pt to improve readability, if needed. You do not have to create assembly code for `printf` (replace directly with assembly instruction). Do NOT debug the C code. HINT: start your translation from `main()`, not line 1.

```
1  float f1(int arr [], int arr_len, int length){
2      float sum = 0;
3      float ma = 0;
4      for (int i=0; i<= (arr_len-length); i++ ) {
5          sum = 0;
6          for ( int j=i; j<i+length; j++ ) {
7              sum += arr[j];
8          }
9          ma = sum / length;
10
11         cout << ma << endl;
12     }
13
14     return ma;
15 }
16
17 int main() {
18     int arr [] = {234,220,219,245,227,238,233,237};
19     int arr_len = sizeof(arr)/sizeof(arr[0]);
20     int length = 3;
21
22     float r1 = f1(arr, arr_len, length);
23
24     if ( r1 > 234 ) {
25         cout << r1;
```

```
26      } else {
27          cout << r1 - 234;
28      }
29
30      return 0;
31  }
```

(b) (4 points) Provide a simplified mathematical expression/equation (without sums or products of series - i.e., no Σ or Π operations, although these may be starting points) that will count the total number of times the code in line 7 will be iterated over in the above f1 function. The expression must be written as a function of an arbitrary array of $n > 0$ items and length variable $m > 0$. Do not compute the output of the function.

(c) (4 points) Create the R equivalent to code in f1 using only existing R function(s). You should need no more than 1 lines of R code to compute the results and 2 lines to print (one for individual values within the for loop, the other to print the final value). You can define the array A and length m on separate lines (not in line count). HINT: use sapply() and don't try to line-by-line convert the C++ code!

(d) (4 points) Consider the following C++ code to compute a weighted moving average, which is often used in financial contexts. Write the associated pseudocode for line 5 as a single, mathematical equation.

```
1  double f2( int A[], int n, float s, int m ) {
2      float mAvg = arr[0];
3
4      for ( int i=1; i<n; i++ ) {
5          mAvg = (arr[i]*(s/(1+m)) + mAvg * (1-(s/(1+m));
6      }
7
8      return mAvg;
9  }
```

(e) (2 points) If using the same variable names from the code in the equation, is it possible to be confused when just looking at the math? If so, correct it.

2. Watch the video at `https://www.youtube.com/watch?v=Xlk4Au0C6TE` and answer the following questions. Note, Shiny is an advanced R package for creating apps.
   (a) (1 point) What is a consequence of making R code more efficient and easier to read?
   (b) (1 point) What are the three characteristics of good variable names?
   (c) (1 point) What is one thing you can do to kindest to yourselves as programmers?
   (d) (1 point) What are two benefits to a vertical layout of text?
   (e) (1 point) What is one reason to be careful about choosing colors to use in plots?
   (f) (1 point) Why is it important to have proper text size on axes, plot titles, etc?
   (g) (1 point) What are the five coding best practices highlighted in this video?

3. Watch the video at `https://www.youtube.com/watch?v=u3YoNpSnmis` and answer the following questions. Note, introductions with short videos to Unit and Integration testing can be found at the following URLs `https://www.guru99.com/unit-testing-guide.html` and `https://www.guru99.com/integration-testing.html`, respectively. While not part of the question it is HIGHLY suggested to read over these two additional links, they will be extremely useful for your project!

(a) (1 point) What should you already have a good grasp of before starting to do large scale software system design?

(b) (1 point) What is one of the more important points on the list of things talked about?

(c) (1 point) Why is it important to define subsystems versus one big code chunk?

(d) (1 point) What are the seven testing approaches/focuses that are highlighted?

(e) (1 point) What should you be thinking about as you write code?

(f) (1 point) Code deployment (moving it from your development environment to its real-world environment) can be tricky, but most people leave until the last minute. What can happen as a result?

(g) (1 point) What was the suggested amount of time spent per day writing documentation for code?

(h) (1 point) Why is scaling something many people do not consider?

(i) (1 point) Why is the runtime complexity of an algorithm important, in a practical sense?

4. (10 points) The project has a number of different components in it. For this question your group will decompose the system into smaller, manageable subsystems that have a specific role and can be independently tested as discussed in Question 3. Additionally, provide a website map (see Step 1 of `https://www.quicksprout.com/creating-website-sitemap/`) and sample website designs (sketches, not HTML) to demonstrate that the group has a vision for what the final product may look like, and how different parts of it will interact.

5. (20 points) This semester your group will be responsible for completing a large coding project, the details of which have been released alongside this assignment. The project is designed to be challenging in order to encourage development of many important skills: teamwork, project management, visualization of results, database design, programming, critical thinking, system design, and more. Completing such a project over many months, while learning the skills to do so, can become quite overwhelming if the group isn't well organized, motivated, determined, helping each other, and holding each other accountable in a transparent way.

   So that your group can start of on the right foot, this question will ask you to clearly outline expectations of each group member so that each is accountable for their efforts, and responsible to the group. Additionally, you will be assigned roles to help organize and ensure a quality final product. Be meticulous in this step as team organization, coordination, and communication can "make or break" a project. **Do not underestimate the importance of this.**

   Provide brief but complete replies to the following questions:

   1. What expectations do you have for each other and the group? Consider at least: group meeting frequency, being on-time and sticking to schedules, weekly hours contributed to the course, participation during meetings (vs just showing up), asking for assistance, timely reply to emails, homework writing process and proof reading, etc.
   2. How will the group manage each member's progress in watching video lectures, and attending the in-person lecture sessions (where a lot of very useful information will be provided to understand the material and complete homework/project, and not posted to Brightspace)?
   3. What is considered "meaningful" contribution to a homework question and to the project? How will the group measure this? How will the group ensure contribution is equal at each homework and final project submission? What should happen to those not equally contributing (be as specific as possible)?
   4. How do you intend to keep track of your own and each others' engagement with the project activities and homework to prevent group members from falling behind on their understanding of the course material and project responsibilities?

5. How will the group go about making important design, coding or other strategic decisions?
6. How will the group handle conflict resolution? (i.e., if one or more members are not contributing sufficiently or fail to attend team meetings, differing opinions to a specific problem, etc.)
7. How will the group conduct a fair procedure for the firing of a non-contributing or disruptive group member?

To help with achieving a high quality final result, assigning specific roles can be very useful. Groups will agree upon quality assurance managers for the following purposes. It is the role of the manager to ensure the group is adhering to quality standards in the different aspects, not necessarily to take on the responsibility to implement those tasks. Below is a suggested set of roles. Each group should have at least one of each.

- **Communications Officer:** responsible for ensuring different members of the group maintain effective communication; documents start/end time of each meeting, who was in attendance, if anybody was late or left early, any members not "meaningfully contributing". Ensures the group is on schedule. This should be kept in an electronic form (will be submitted as requested throughout the semester, so be sure this is accurate and up to date!).
- **Code Quality Officer:** responsible for ensuring the group maintains and adheres to rules for how to name functions, variables, files; that each member is adhering to these, and that different code versions are properly monitored; identifying inefficient code. The rules should be smartly decided upon by the group to enforce good code.
- **Documentation Officer:** responsible for ensuring code is properly documented, that plots/visualizations are easily readable, etc.
- **Report Quality Officer:** responsible for ensuring that each member reads all submitted homework and reports, provides input to its language, editing, etc.,
- **Contribution Officer:** responsible for ensuring the group is monitoring "meaningful" contribution to each homework and the project as a whole in a transparent way. Reports concerns to the TAs or instructors.
- **System Testing Officer:** responsible for ensuring that the group is constantly testing functions and the overall software system in an effective way. NOTE: this includes testing any user interfaces and webpage functionality as that is the product of code.

Think of these roles primarily as the questions one would be asking of the group during meetings. For instance, suppose the group is addressing a homework question that has a coding aspect to it. The following would be SOME of the questions each member could bring up:

- **Communications Officer:** Let's start the meeting, who is here? Is the question done on time?And at the end of the meting, who were the members not equally contributing at this meeting, nor completed their task from the last meeting (and why not)?
- **Code Quality Officer:** Does the code have proper variable names, is it properly split into different functions? Is it efficient code?
- **Documentation Officer:** Was the code properly documented? Are the output plots easily readable? How do we know the result is correct?
- **Report Quality Officer:** Does anybody have any edits to how this is written in the homework solution? Is this too long/confusing to read?
- **Contribution Officer:** Is everybody contributing in an equally meaningful way? If not, why? How can we fix this?
- **System Testing Officer:** Are we sure the code is properly working? Did we test it well?