School of Engineering and Information Technology

ASSESSMENT COVER SHEET

Student Name	Shane Reynolds
Student ID	262538
Assessment Title	Quiz 2
Unit Number and Title	HIT365
Lecturer/Tutor	Kai Wang
Date Submitted	18/4/2014
Date Received	18/4/2014

KEEP A COPY

Please be sure to make a copy of your work. If you have submitted assessment work electronically make sure you have a backup copy.

PLAGIARISM

Plagiarism is the presentation of the work of another without acknowledgement. Students may use a limited amount of information and ideas expressed by others but this use must be identified by appropriate referencing.

CONSEQUENCES OF PLAGIARISM

Plagiarism is misconduct as defined under the Student Conduct By-Laws. The penalties associated with plagiarism are designed to impose sanctions on offenders that reflect the seriousness of the University's commitment to academic integrity.

I declare that all material in this assessment is my own work except where there is a clear acknowledgement and reference to the work of others. I have read the University's Academic and Scientific Misconduct Policy and understand its implications.*

http://www.cdu.edu.au/governance/documents/AcademicandScientificMisconductPolicyv1.03Jan2011.pdf.

Signed……  ……Date……18/4/2014……

*** By submitting this assignment and cover sheet electronically, in whatever form you are deemed to have made the declaration set out above.**

C PROGRAMMING – QUIZ 2

HIT 365 – C Programming

Q1 An array called `marks` has been created in a C program of length ASIZE. There are two values out of numerical order in the array and these values are stored at indexes 0, and 7. Write code to swap those two values so that the array would be in order.

Response

```
/*
 * Question1.c
 * Created on: 18/04/2014
 * Author: Shane
 */

#include <stdio.h>
#define ASIZE 7

/* Function prototype */
void printArray(int array[ASIZE]);

int main(void){

	/* Initialise variables */
	int temp;
	int marks[ASIZE] = {97, 65, 68, 73, 75, 83, 63};

	/* Print original array */
	printArray(marks);
	printf("\n\n");

	/* Swap variables using a temp storage variable */
	temp = marks[0]; /* Stores first element of array in temp */
	marks[0] = marks[6]; /* Assigns last element to first */
	marks[6] = temp; /* Put the first element in last */

	/* Prints the updated array */
	printArray(marks);
	printf("\n\n");

	return 0;
}

/* Function prints an array of static size ASIZE */
void printArray(int array[ASIZE]){

	int i;

	for(i = 0; i < ASIZE; i++){
			printf("%d ", array[i]);
		}

}
```

Q2 The following questions refer to a character array called `fantasy`. You do not know what values are stored in `fantasy`, and you do not how many elements `fantasy` contains, except there is at least one.

(a) In the box below, write C code to print the first element of the array

Response

```
/* Answer to part a */
printf("%c\n\n", fantasy[0]);
```

(b) In the box below, write C code to print the final element of the array

Response

```
/* Answer to part b */
printf("%c\n", fantasy[sizeof(fantasy)-2]);
```

(c) The code block below operates on the array `fantasy`. The variable `index` contains an unknown value but it is known that: `0 ≤ index <` the length of the array `fantasy`.

The code prints out the value of the array element one place to the ***left*** of `fantasy[index]`, but if there is no element to the left of `fantasy[index]`, it prints an appropriate error message:

```
if (index > 0) {
   printf("%c",fantasy[index - 1]);
} else {
   printf("nothing to the left");
}
```

In the box below, write C code to print out the value of the array element one place to the ***right*** of `fantasy[i]`, but if there is no element to the right of `fantasy[i]`, it prints an appropriate error message. As for the code above, it is known that: `0 ≤ index <` the length of the array `fantasy`.

Response

```
/* Answer to part c */
for (i = 0; i < sizeof(fantasy); i++){
      if ( i < (sizeof(fantasy)-1) ){
            printf("%c", fantasy[i+1]);
      } else {
            printf("Nothing to the right.");
      }
}
```

(d) Write an `if` statement that compares the values in `fantasy` at `index` and the value in `fantasy` one place to the right, and prints out the larger of those two values (or either of them if they are equal). It is known that: `0 ≤ index <` one less than the length of the array `fantasy` so your code should not test for that.

Response

```
/* Answer to part d */
for (i = 0; i < sizeof(fantasy); i++){

	if ( i < (sizeof(fantasy)-1) ){

		if(fantasy[i] > fantasy[i+1]){
			printf("%c", fantasy[i]);
		} else {
			printf("%c", fantasy[i+1]);
		}

	} else {

		printf("\n\nNothing to the right.");

	}
}
```

Q3 Consider the following block of C code:

```
if (num <= 0) {

    printf("A");

}

if (num > 3) {

    printf("B");

}

if (num % 2 == 0) {

    printf("C");

}
```

(*The `%` operator finds the remainder after division. For example: 9 % 4 would return 1*)

Provide a value for `num` which would cause 'C' (and no other letter) to be printed.

Response

Let `num` equal 2.

Q4 This question refers to the following code, where the variables `p`, `q`, `r` and `s` all have integer values:

```
if (p < q) {
    if (q > 4) {
        s = 5;
    } else {
        s = 6;
    }
}
```

Assume that, **before** the above code is executed, the values in the four variables are:

```
int p = 4;  int q = 5;  int r = 3;  int s = 4;
```

What would be the value in variable `s` after the code is executed?

Response

The value of `s` is 5.

Q5 Consider the following block of code, where variables `a`, `b` and `c` each store integer values:

```
if (a > b) {
    if (b > c) {
        printf("%d\n", c);
    } else {
        printf("%d\n", b);
    }
} else if (a > c) {
    printf("%d\n", c);
} else {
    printf("%d\n", a);
}
```

(a) In relation to the above block of code, which one of the following values for the variables will cause the value in variable `b` to be printed?

		Response
(i)	a = 1; b = 2; c = 4;	`b` will not print.
(ii)	a = 1; b = 4; c = 2;	`b` will not print.
(iii)	a = 2; b = 1; c = 8;	`b` will print.
(iv)	a = 4; b = 2; c = 1;	`b` will not print.

(b) In one sentence that you should write in the box below, describe the purpose of the above code (i.e. the `if/else if/else` block). Do **NOT** give a line-by-line description of what the code does. Instead, tell us the purpose of the code:

Response

The code finds the minimum value for given values of a, b and c.