Université d'Ottawa
Faculté de genie

École d'ingénierie et de
technologies de l'information

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Information
Technology and Engineering

GNG1106 A/B Fall 2013 Final Exam

- Three Hours
- Closed book
- Non-programmable calculators are allowed.
- The full mark of this paper is 100 points. But you may get up to 104+4=108 points. That is, you may get up to 8 points bonus.
- Do not write on the exam paper. Only answers written in the exam booklets are considered for marking.

Part 1 (5 points × 7 = 35 points)

Questions 1-10 are multiple-choice questions. Except Question 6 which may have multiple correct answers, every question has only one correct answer. You do not need to justify your answer and, if you do, your justification will be ignored in marking.

Question 1 *Suppose that variable* `x` *is declared as* `int` *type. Which one of the following statements will read the value of x from the keyboard?*

(A) `scanf("%d", x);`

(B) `scanf("%d", &x);`

(C) `printf("%d", x);`

(D) `printf("%d", &x);`

Question 2 *What will be the value of variable* `i` *after the following code segment is executed?*

```
int x=0;
int i;
for (i=0;i<100;i++)
{
        x=x+i;
        if (x>20)
                break;
}
```

(A) 5

(B) 6

(C) 7

(D) None of above.

Question 3 *Regarding the following program, which one of the following statements is correct?*

```
#include <stdio.h>


int foo(int a, int b)
{
        int c=0;
        c=c+b-a;
        a=b;
        return c;
}

int main(void)
```

```
{
        int a=1;
        int b=2;
        int c=3;
        c=foo(a, b);
        printf("a=%d, c=%d\n", a, c);
        return 0;
}
```

(A) It prints "a=2, c=1"

(B) It prints "a=1, c=1"

(C) It prints "a=2, c=3"

(D) It prints "a=1, c=3"

Question 4 *Regarding the following program, which one of the following statements is correct?*

```
#include <stdio.h>


int foo(int *a, int b)
{
        int c=0;
        c=c+b-*a;
        *a=b;
        return c;
}

int main(void)
{
        int a=1;
        int b=2;
        int c=3;
        c=foo(&a, b);
        printf("a=%d, c=%d\n", a, c);
        return 0;
}
```

(A) It prints "a=2, c=1"

(B) It prints "a=1, c=1"

(C) It prints "a=2, c=3"

(D) It prints "a=1, c=3"

(E) None of above is correct

Question 5 *Regarding the following program, which one of the following statements is correct?*

```
#include <stdio.h>

int foo(int a, int b)
{
        static int c=0;
        c=c+b-a;
        a=b;
        return c;
}

int main(void)
{
        int a=1;
        int b=2;
        int c=3;
        c=foo(a, b);
        c=foo(a, b);
        c=foo(a, b);
        printf("c=%d\n",  c);
        return 0;
}
```

(A) It prints "c=1"

(B) It prints "c=2"

(C) It prints "c=3"

(D) It prints "c=4"

(E) None of above is correct

Question 6 *An array* `a` *is declared by*

```
int a[10];
```

In the expressions below, which ones are equivalent to the expression `a[5]`*?*

(A) `*(a+5)`

(B) `&(a+5)`

(C) `&(*a[5])`

(D) `*a+5`

(E) `&(*a+5)`

Question 7 *In order to use function* `malloc`*, which one of the following lines should be contained in your code?*

(A) `#include <stdio.h>`

(B) `#include <stdlib.h>`

(C) `#include <string.h>`

(D) None of above

Part 2 (8 points × 3 = 24 points)

In Questions 8-10, feel free to justify your answers, which will be considered in marking.

Question 8 *The C code of a program is given below. How can you run this program so that what the program does is precisely printing to the screen the string "*`I LIKE EXAMS`*"?*

```
#include <stdio.h>

int main(int argc, char **argv)
{
        if (argc>8)
                printf("%s %s %s\n", argv[2], argv[4], argv[6]);
        return 0;
}
```

Question 9 *What is the execution result of the following program?*

```
#include <stdio.h>

int foo(int *a, int b)
{
        static int c=0;
        c=c+b-*a;
        *a=b;
        printf("in foo: *a=%d, b=%d, c=%d\n", *a, b, c);
        return c;
}

int main(void)
{
        int a=1;
        int b=2;
        int c=3;
        c=foo(&a, b);
        printf("in main: a=%d, b=%d, c=%d\n", a, b, c);
        c=foo(&c, b);
        printf("in main: a=%d, b=%d, c=%d\n", a, b, c);
        return 0;
}
```

Question 10 *A program is given below (where the statements for including the correct header files are not shown).*

```
void func1(int *b)
{
        int *a;
        a=(int *)malloc(3*sizeof(int));
        a[0]=1;
```

```
        a[1]=2;
        a[2]=3;
        b=a;
}


void func2(int **b)
{
        int *a;
        a=(int *)malloc(3*sizeof(int));
        a[0]=-1;
        a[1]=-2;
        a[2]=-3;
        *b=a;
}


int main(void)
{

        int *p;
        int i;
        p=(int *)malloc(3*sizeof(int));
        p[0]=11;
        p[1]=12;
        p[2]=13;
        func1(p);
        for (i=0;i<3;i++)
                printf("%d\n", p[i]);

        func2(&p);
        for (i=0;i<3;i++)
                printf("%d\n", p[i]);

        return 0;

}
```

What is the execution result of the code?

Part 3 (45 points)

Questions 11-14 are programming questions. Partial marks will be given to your code and to your demonstration of correct and **relevant** logic.

Question 11 (9 points) *You are to write a function with prototype* `float func(void)`. *The function, when called, will keep prompting the user to enter* **positive** *float numbers until the user enters a negative number. The function discards the largest number and the smallest positive numbers that the user has entered and returns the average of the remaining positive numbers. (***You may get an additional 4 bonus points if your function uses the minimal amount of memory and takes the minimal amount of time.***)**

Question 12 (9 points) *A text file contains 1000 float-type numbers in the range from -1000 to 1000. Write a program that checks how many numbers in the file that is in the range from -1 to 1.*

Question 13 (15 points) *Two structures* `struct car` *and* `struct carList` *are defined as below.*

```
struct car
{
        char make[30],
        char model[30],
        int year;
        int mileage;
};

struct carList
{
        int numberOfCars;
        struct car *cars;
};
```

The structure `struct car` *is meant to encapsulate some basic information of a car, and the structure* `carList` *is meant to encapsulate a list of cars.*

1. *Write a function with prototype* `void printCar(struct car myCar)` *which prints to the screen the make, model and year of a car passed to the function via variable* `myCar`.
2. *Write a function with prototype* `void printCarList(struct carList myList)` *which prints the make, model and year of every car in* `myList`.
3. *Write a function with prototype* `struct carList *findCarsBasedOnYear(int yearOfInterest, struct carList *myList)` *which searches the list of cars specified by pointer* `myList` *according to the year specified by* `yearOfInterest` *and returns a list of cars via the returned pointer to* `struct carList`. *More precisely, the returned list of cars contains all cars in* `myList` *whose year is* `yearOfInterest`.

Question 14 (12 points) *The following code segment defines a linked list, where each node contains a pointer field and a data field of* `int` *type.*

```
struct node
{
        int data;
        struct node *next;
};

struct linkedList
{
        struct node *head;
};
```

1. Build a function with prototype `void appendToTail(linkedList *theList, int x)` that creates a new node whose `data` field has value `x` and appends the node to the tail of the linkedList whose address is `theList`.
2. Build a function with prototype `node *isXInLinkedList(linkedList *theList, int x)` that checks if any node in the linked list whose address is `theList` contains a node whose `data` field has value `x`. The function returns the address of the found node. It returns NULL if no such node is found.