hw1: C Programming 1

Due Feb 13 at 11:59pm **Points** 8 **Questions** 8
Available Feb 7 at 12am - Feb 14 at 11:59pm **Time Limit** 40 Minutes
Allowed Attempts 2

Instructions

You may study between attempts and use your notes for other quiz attempts. Exams will not have have multiple attempts or allow use of notes.

You will be able to see your quiz score but you will not see correct answers until after quiz LATE period has passed. Later attempts may have different questions.

There is a 10% penalty for LATE submissions. Submit your last attempt before the DUE date to avoid this penalty.

Homework Quizzes cover the previous lectures, and related reading assignments

- C Programming
- C Program Structure and Control Flow
- Arrays and Pointers

You may retake this quiz after reviewing the information that you did not know during the quiz. Multiple attempts are allowed for most quizzes, but there is only one attempt for each exam.

Once you submit your attempt, you will not be able to review the questions or answers in the attempt until results are released. Results are only released after the due date and time has passed for all students. Do not post questions about the quiz on piazza. Save your questions the quiz results have been released and then only post quiz questions privately to piazza.

This quiz was locked Feb 14 at 11:59pm.

Attempt History

	Attempt	Time	Score
KEPT	Attempt 2	17 minutes	8 out of 8
LATEST	Attempt 2	17 minutes	8 out of 8
	Attempt 1	40 minutes	4 out of 8

Attempt	Time	Score

Score for this attempt: **8** out of 8

Submitted Feb 13 at 6:26pm

This attempt took 17 minutes.

Question 1

1 / 1 pts

```
#include <stdio.h>

int main(void) {
  int i = 11;
  int *ptr1 = &i;
  int **ptr2 = &ptr1;
  int ***ptr3 = &ptr2;

  printf("%p, %p, %p, %p\n", ptr1, ptr2, ptr3, &ptr3);

  return 0;
}
```

If the program output is (where _ is part of the address that is not shown):

```
0x_0b4, 0x_0b8, 0x_0c0, 0x_0c8
```

- At address `0x_0b4` is stored [Select] .
- At address `0x_0c8` is stored [Select] .

Answer 1:

Correct!

11

Answer 2:

Correct!

0x_0c0

Question 2

1 / 1 pts

Consider the following code:

```
#include <stdio.h>
#include <string.h>

int main(void) {
        char destination[16] = "batman";
        char source[] = "spiderman";
        strcat(destination, source);
        printf("%s %s %d %d", destination, source, sizeof(destinatio
n), sizeof(source));
        return 0;
}
```

The program output is?

○ `spiderman spiderman 7 10`

○ `batmanspiderman spiderman 15 9`

○ `spiderman spiderman 10 10`

Correct!

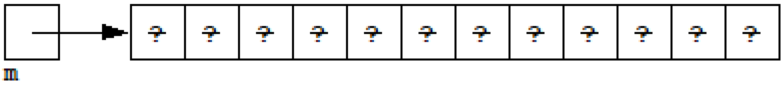
● `batmanspiderman spiderman 16 10`

○ `batmanspiderman spiderman 7 10`

Question 3

1 / 1 pts

Consider the following memory diagram where `m` **is on the stack** and the rest is heap memory. It is intended to be used as a 2-dimensional matrix of integers having 3 rows with each having 4 columns (where ? indicates an uninitialized integer value):



Which one of the following code fragments will allocate the heap memory as diagrammed above?

○
```
int **m = malloc(sizeof(int*) * 3);
m[0] = malloc(sizeof(int) * 3 * 4);
m[1] = m[0] + 4;
m[2] = m[1] + 4;
```

○
```
int *m = malloc(sizeof(int*) * 3);
m[0] = malloc(sizeof(int) * 3 * 4);
```

Correct!

●
```
int *m = malloc(sizeof(int) * 3 * 4);
```

○
```
int **m = malloc(sizeof(int*) * 3 * 4);
```

○
```
int **m = malloc(sizeof(int*) * 3);
m[0] = malloc(sizeof(int) * 4);
m[1] = malloc(sizeof(int) * 4);
m[2] = malloc(sizeof(int) * 4);
```

Question 4

1 / 1 pts

Consider the following code:

```
#include <stdio.h>

void update(int *x) {
    int *a;
    CODE A
    printf("%d ", *a);
}

int main(void) {
    int a = 4;
    int *b = &a;
    printf("%d ", *b);
    update(b);
    printf("%d ", *b);
    return 0;
}
```

Which one of the following is FALSE?

if CODE A is:

```
a = malloc(sizeof(int));

*a = 1;

x = a;
```

then the output is:

○ `4 1 4`

if CODE A is:

```
a = x;

*a = 16;
```

then the output is:

○ `4 16 16`

if CODE A is:

```
a = malloc(sizeof(int));

a = x;

*x = 1;
```

then the output is:

○ `4 1 1`

Correct!

if CODE A is:

```
*a = 16;
```

then the output is:

● `4 16 4`

Question 5

1 / 1 pts

Consider the following code:

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    int *x = malloc(sizeof(int) * 5);
    for(int i = 0; i < 5; i++) {
        *(x+i) = 4 - i;
    }
    CODE A
    return 0;
}
```

Assume that size of an integer is 4 bytes and value of x to be 1000 in decimal. Which of the the following 4 cases for CODE A are correct?

1. If CODE A is:

```
int *y = x+1;
printf("%d %d %d", x, y, y-x);
```

Output will be: `1000 1004 1`

2. If CODE A is:

```
printf("%d ", *x);
x += 2;
printf("%d", *x);
```

Output will be: `0 2`

3. If CODE A is:

```
int *y = x + 4;
printf("%d %d %d" , x, y, *(x+*y));
```

Output will be: `1000 1016 4`

4. If CODE A is:

```
int *y = x + 1;
printf("%d %d %d" , x, y, y-x);
```

Output will be: `1000 1008 2`

Correct!

● 1 and 3

○ 1, 2, 3, and 4

○ 1 only

○ 2 and 4

○ 2 only

Question 6

1 / 1 pts

```
#include <stdio.h>

void f(int num1, int num2, int *ptr3) {
        int temp = *(ptr3 + num1);
        ptr3[num1] = ptr3[num2];
        *(ptr3 + num2) = temp;
}

int main(void) {
        int a = 3;
        int b = 0;
        int c[] = {13, 3, 21, 8, 2, 5};

        f(a, b, c+1);

        printf("%i,%i,%i,%i,%i,%i\n", c[0], c[1], c[2], c[3], c[4], c
[5]);

        return 0;
}
```

Which one of the following shows the output of the program?

○ 8,3,21,13,2,5

○ 13,3,21,8,2,5

Correct! ● 13,2,21,8,3,5

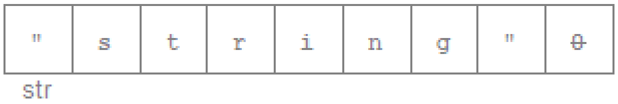
○ 13,3,5,8,2,21

Question 7

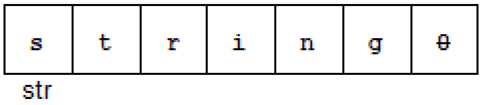
1 / 1 pts

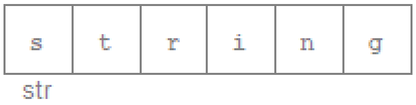
Which one of the following diagrams the memory corresponding to this code:

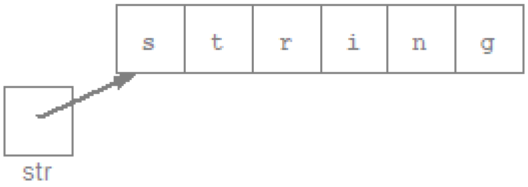
```
char str[] = "string";
```

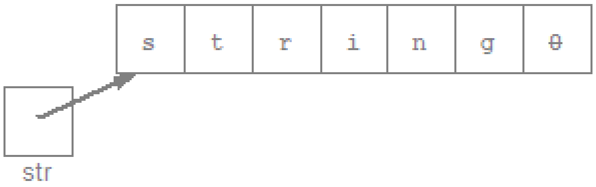
○ 

Correct!

● 

○ 

○ 

○ 

Question 8

1 / 1 pts

Consider the following code:

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    int *x[5];

    for(int i = 0; i < 5; i++) {
        x[i] = malloc(sizeof(int) * 5);
    }
    for(int i = 0; i < 5; i++) {
        for(int j = 0; j < 5; j++) {
            x[i][j] = i * j;
        }
    }
    modify(x, 5, 5);
    return 0;
}
```

Which of the implementations of method `modify` below set all elements of the matrix `x` to zero?

```
1. void modify(int **x, int m, int n) {
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                x[i][j] = 0;
            }
        }
   }
2. void modify(int *x[], int m, int n) {
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                x[i][j] = 0;
            }
        }
   }
3. void modify(int x[5][5], int m, int n) {
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                x[i][j] = 0;
            }
        }
   }
```

○ 1, 2 and 3

○ 2 only

○ 1 and 3

Correct! ● 1 and 2

○ 2 and 3

Quiz Score: **8** out of 8