Homework 1

- Due Sep 24 at 11:59pm
- Points 100
- Questions 25
- Available Sep 17 at 9am - Sep 24 at 11:59pm
- Time Limit None

Instructions

While this HW is due after Exam 1, I strongly suggest you complete this before the exam. These are SOME of the concepts that may the on Exam 1.

This quiz was locked Sep 24 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	108 minutes	96 out of 100

Score for this quiz: 96 out of 100

Submitted Sep 22 at 3:31pm

This attempt took 108 minutes.

Question 1

2 / 4 pts

Create a struct that has variables that represent the following attributes of a car:

 The make of a car.

The model of a car.

The color of the car.

The year of the car.

I should be able to create an instance in the following way:

struct car;

car_t:

Your Answer:

```
struct car {

char make[];

char model[];

char color[];

int year;

};
```

⣿

Question 2

4 / 4 pts

Consider the following structs:

```
struct simple{

   int a;

   char b;
   float c;
};
struct Init{
   int a
   short b[10]
   struct simple c;
};
```

Rewrite struct Init, such that during the creation of the struct it will initialize an instance of Init using an initialization list.

Your Answer:

```
struct Init myInit = {
.a = 42, // Initialize 'a' to 42
.b = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, // Initialize 'b' array with values
.c = {5, 'z', 3.14f} // Initialize the 'c' member (a struct simple)
}
```

Question 3
4 / 4 pts

Consider the following struct:

```
struct  test{
   int a
   struct test b;
   char c;
};
```

This struct is broken. Explain why it is broken and how to fix it.

Your Answer:

Inside of the test struct, we are defining a struct test that contains another instance of itself which leads to infinite recursion. To fix this, we make a pointer to struct test. There is also a missing semi-colon after int a.

```
struct test {
int a;
struct test* b;
char c;
};
```

Question 4
4 / 4 pts

The following struct is broken. Describe the problem and how to fix the problem.

```
typedef struct {
   char a

   struct SELF_REF *b;

   int c;

}SELF_REF;
```

Your Answer:

Missing semi-colon after char a, also to define SELF_REF, it needs to be properly introduced beforehand.

```
typedef struct SELF_REF {

char a;

struct SELF_REF* b;

int c;

} SELF_REF;
```

Question 5
4 / 4 pts

Consider the following struct:

```
struct First

{

   short a;

   float b;
   char c;

   int d;

   int e;

};
```

Answer the following questions:

What is the largest data type in this struct? Your answer should be a number only. 4

How many bytes of memory will be needed for this struct? Your answer should be a number only. 20

How many bytes of padding will be needed for this struct? Your answer should be a number only. 5

Answer 1:

Correct! 4

Correct!

4

Answer 2:

Correct! 20

Correct!

20

Answer 3:

Correct! 5

Correct!

5

Question 6
4 / 4 pts

This question is a follow up of the previous question.

Consider the following struct.

```
struct First

{

   short a;

   float b;

   char c;

   int d;

   int e;

};
```

Rewrite this struct using the rule discussed in class that will minimize the number of bytes needed and push the padding to the end of the struct.

Your Answer:

```
struct First
{
float b;

int d;

int e;

short a;
char c;

}
```

⁝⁝

Question 7

4 / 4 pts

In class we discussed the following program.

```
#include <stdio.h>
struct date {
int d : 5;
int m : 4;
int y;
};

int main()
{
struct date dt = { 31, 12, 2020 };
printf("Date is %d/%d/%d\n", dt.d, dt.m, dt.y);
printf("%lu", sizeof(struct date));
return 0;
}
```

This program has a problem. When I run the program I get the following output:

```
Date is -1/-4/2020
```

Describe why his program does not work properly and how to fix the problem.

Your Answer:

Since we are signing int d and int m to 5 and 4 respectively, the most significant (left most) bit is treated as a sign bit. In other words, the leading 1 indicates a negative value. The range of values for d : 5 is from -16 to 15, and since 31 exceeds this range, it overflows. This is the same with m : 4 when given 12 in a range of -8 to 7.

To fix this we need to use unsigned for the bit fields.

struct date {

unsigned d : 5;

unsigned m : 4;

int y;

};

Question 8

4 / 4 pts

I can use bit fields with a floating point number.

○ True

Correct!

[✓ False]

Question 9

4 / 4 pts

With respect to bit fields:

We cannot create an array of bit fields.

We can print the address of a bit field.

○ True

Correct!

[✓ False]

Question 10

4 / 4 pts

As described in the notes, what is a pointer? Your answer should have two answers.

A pointer is:

In programming a pointer is:

Your Answer:

A pointer is something that points to something else.

In programming, a pointer is a data type that holds the address of another variable.

Question 11

4 / 4 pts

In class and in the notes, there are several reasons listed for why we would use pointers. List 3 of the reasons.

Your Answer:

Gives the ability to work with particular memory locations.

Allows us to effectively represent complex data structures.

Optimization of programs - passing a pointer to a function does not have to duplicate the data.

Question 12

4 / 4 pts

Consider the following program. This program will print three values what are the values?

```
#include <stdio.h>

int main()
{
int arr[ 10 ] = {10,19,45,18,13,98,200,33,99,1};
int * ptr = arr;
printf("value of what ptr is pointing to: %d\n", *ptr);
int* ptr2 = ptr+5;
printf("value of what ptr2+5 is pointing to: %d\n", *(ptr2));
printf("value of what (ptr + 5) +2 is pointing to: %d\n", *(ptr+5)+2);

return 0;
}
```

Print statement 1: value of what ptr is pointer to: 10

Print statement 2: value of what ptr2+5 is pointing to: 98

Print statement 3: value of what (ptr+ 5) + 2 is pointing to: 100

Answer 1:

Correct! 10

Correct!

10

Answer 2:

Correct! 98

Correct!

98

Answer 3:

Correct! 100

Correct!

100

⋮⋮

Question 13

4 / 4 pts

What will print for each print statement?

```
#include <stdio.h>


int main()
{
   int iArr[] = {0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20};
   int *iPtr = iArr;

   printf("%d", iPtr[ 0 ]);          0

   printf("%d", iPtr[ 0 ] +1);       1

   printf("%d", *(iPtr+5));          10

   printf("%d", *(iPtr+3)+5);        11
```

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

4

```
return 0;
}
```

Answer 1:

Correct! 0

Correct!

0

Correct Answer

O

Answer 2:

Correct! 1

Correct!

1

Answer 3:

Correct! 10

Correct!

10

Answer 4:

Correct! 11

Correct!

11

Answer 5:

Correct! 4

Correct!

4

Question 14

4 / 4 pts

What is the output of the following program?

```
#include <stdio.h>

int f(int , int *, int **);

int main()
{
int c, *b, **a;
c = 15;
b = &c;
a = &b;

printf("%d \n", f(c, b, a));
return 0;
}

int f(int x, int *py, int **ppz)
{
int y, z;
(*(*ppz)) += 1;
z = **ppz;
*py += 5;
y = *py;
x += 2;
return x + y + z;
}
```

Correct!

54

Correct Answers

54

⋮⋮

Question 15

4 / 4 pts

In class we discussed two differences in the C provided functions malloc and calloc. Describe the differences?

Your Answer:

Calloc initializes the data to zero, where as malloc does not and contains garbage values.

Malloc takes one arg which is the total number of bytes to allocate.

Calloc takes two args, one being the number of elements to allocate, and the second as the size of each element.

⋮⋮

Question 16

4 / 4 pts

This is a problem that involves dynamically allocating memory in 1 dimension but accessing the data as if it were 2 dimensional.

Assume the following variables are created and initialized.

int row = 3; int col = 4;

int *data = malloc(row * col * sizeof(int));

int i, j, count = 0;

Fill in the appropriate information in the nested for loops to initialize the dynamically allocated memory for data:

for(i = 0; i < row; i++)

```
{
    for(j = 0; j < col; j++)
   {
        data[ i * col + j ] =  ++count;
   }
```

What is the code to give the dynamically allocated memory back to the operating system? free(data)

Answer 1:

Correct! i * col + j

Correct Answer

i*col+j

Correct!

i * col + j

Answer 2:

You Answered free(data)

Correct Answer

free(data);

Correct Answer

free(data);

Question 17

4 / 4 pts

THIS IS A 2 PART QUESTION:

PART 1: Write the code to dynamically allocate memory for a 2D array of integers. Hint think about the skit with the student pointers, pointing to groups of student data.

PART 2: Write the code to give the dynamically allocated memory back to the operating system.

Your Answer:

PART 1

```
int main() {

int rows = 3;

int col = 4;

int **arr = (int **)malloc(rows * sizeof(int *));

for (int i = 0; i < rows; i++) {

arr[i] = (int *)malloc(cols * sizeof(int));

}
```

PART 2

```
for (int i = 0; i < rows; i++) {

free(arr[i])

}

free(arr)

return 0;

}
```

Question 18

4 / 4 pts

THIS IS A 2 PART QUESTION:

PART 1: Write the code to dynamically allocate memory for a 2D array of integers using the algorithm discussed that requires only 2 calls to malloc or calloc.

PART 2: Write the code to give the memory back to the operating system.

Your Answer:

PART 1

```
int main() {

int rows = 3;

int cols = 4;

int **arr = (int **)malloc(rows * sizeof(int));

arr[0] = (int *)malloc(rows * cols * sizeof(int));


for (int i = 0; i < rows; i++) {

arr[i] = arr[0] + i + cols;

}
```

PART 2

```
free(arr[0]);

free(arr);

return 0;

}
```

Question 19
4 / 4 pts

THIS IS A 2 PART QUESTION:

PART 1: Write the code to dynamically allocate memory for a 2D array of integers using the algorithm discussed that requires only 1 calls to malloc or calloc.

PART 2: Write the code to give the memory back to the operating system.

Your Answer:

PART 1

```
int main() {

int rows = 3;

int cols = 4;


int *arr = (int *)malloc(rows * cols * sizeof(int));

if (arr = NULL) {

return 1;

}
```

PART 2

free(arr);

return 0;

}

Question 20

4 / 4 pts

This code does not swap the values of an and b. Explain why it does not work. Using the line numbers explain how to fix this code such that when it is ran it will swap the value of a and b.

```
1.  void swap(int , int );

2.  int main(){
3.  int a = 5;
4.  int b = 10;
5.  swap(a, b);
6.  printf("a = %d b = %d\n", a, b);
7.  return 0;
}
8.    void swap(int x, int y){
9.    int temp = x;
10.  x = y;
11.  y = temp;
12.  }
```

Your Answer:

Firstly, since we are not taking the address of a or b, when we use swap the values are correctly swapped but the values are just the duplicates of a and b and not their addresses. In other words, we are only swapping their duplicate values and not their addresses themselves.

```
void swap(int *, int *) // this is so the function accepts pointers


int main() {

int a = 5;

int b = 10;

swap(&a, &b);

print("a = %d b = %d\n", a, b);

return 0;

}


void swap(int *x, int *y) {

int temp = *x;

*x = *y;

*y = temp;

}
```

Question 21
4 / 4 pts

Both of the C functions, fread and fwrite, read and write data one character at a time?

○ True

Correct!

[✓ False]

Question 22

4 / 4 pts

The C function rewind moves a specific file pointer to the beginning of the file and returns 0 or 1 to indicate the success of the function.

○ True

Correct!

[✓ False]

Question 23

4 / 4 pts

Given the following function prototype:

int fseek(FILE* str, long int offset, int pos);

Describe what each of the following parameters represent:

str: the file pointer to th

offset: the amount of bytes

pos: the position of whicl

Answer 1:

You Answered the file pointer to the file that will be changed

Correct Answer

the opened file pointer

Answer 2:

You Answered the amount of bytes to move the file pointer

Correct Answer

the number of bytes or characters - determines where the position indicator needs to be place to define the new position of the file pointer

Answer 3:

You Answered the position of which to add the bytes

Correct Answer

defines the point where the ice offset needs to be added; basically it defines the position where the file pointer must be moved

Correct Answer

defines the position where the file pointer must be moved

Question 24

4 / 4 pts

Name and describe the three constants mostly used with fseek:

Your Answer:

SEEK_END: The end of the file.

SEEK_SET: The start of the file.

SEEK_CUR: The file pointer's current position

Question 25

2 / 4 pts

This is a 2 part question:

If given the following set of files:

driver.c, functions.c, functions.h, data.txt

1. Write the command needed to tar these files. Your tar file should be named: HW1TarExample.tar.gz

2. Write the command needed to untar the files.

Your Answer:

tar:

-czvf HW1TarExample.tar.gz driver.c functions.c functions. h data.txt

untar:

-xzvf HW1TarExample.tar.gz

Quiz Score: 96 out of 100