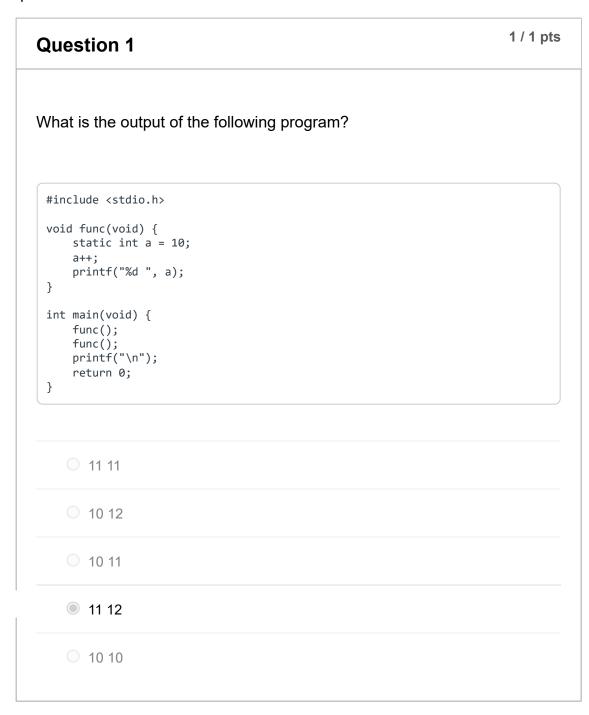
# hw2: C Programming 2 and Virtual Address Space Results for VARDAAN KAPOOR (He/him)

Score for this attempt: **8** out of 8 Submitted Feb 20 at 11:39pm This attempt took 39 minutes.

Correct!



Question 2 1/1 pts

To avoid a limitation in Canvas, array code below has an extra space before the index.

```
typedef struct {
    char name[ 11];
    char *type;
    float weight;
} Pokemon;

int main(void) {
    Pokemon pokedex[ 7];
```

The description of pokedex is an array of Pokemon structures?

The description of pokedex [ 3] is a Pokemon structure?

# Answer 1:

Correct!

an array of Pokemon structures

### Answer 2:

Correct!

a Pokemon structure

Question 3

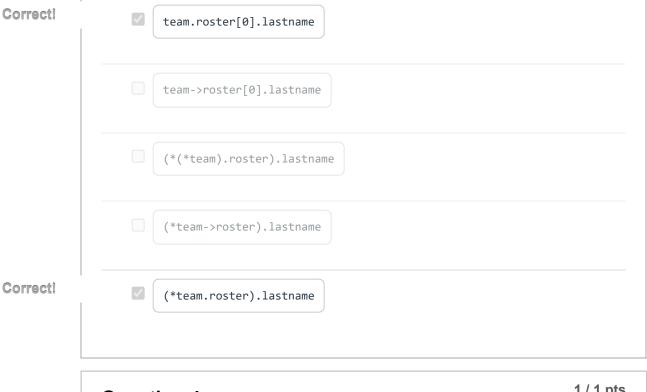
```
typedef struct {
    char firstname[22];
    char lastname[22];
    char position[22];
} Teammate;

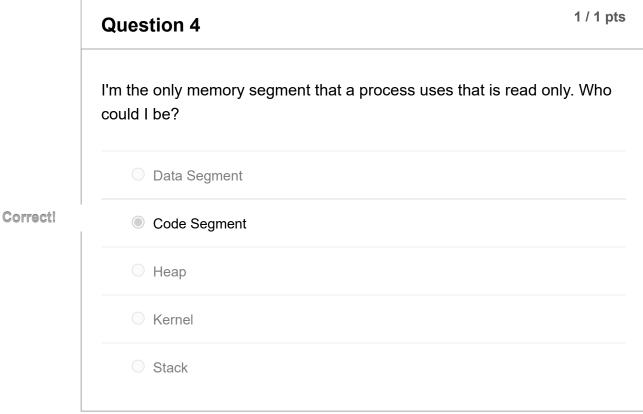
typedef struct {
    int size;
    char name[22];
    Teammate roster[33];
} Team;
```

Assume team has been initialized as follows:

```
Team team;
```

Which of the following code fragments will access the lastname data member of the teammate at index 0. Select all that are correct.





```
#include <stdlib.h>
#include <string.h>
typedef struct {
```

```
char title[55];
    char *author;
    int
        pages;
} Book;
typedef struct {
        numBooks;
    int
    Book *books[22];
} Shelf;
int main(void) {
    Shelf bookShelf[11];
    Book book;
    bookShelf[3].books[7] = NULL;
    //statement(s) added here
    strcpy(bookShelf[3].books[7]->author, "Seuss");
```

Which of the following statements are required to allocate heap memory so that the last statement will make "Seuss" the author of a book in the bookshelf? Select all that are required and don't worry about their order if more that one statement is selected.

# Correct!

Correct!

```
bookShelf[3].books[7] = malloc(sizeof(Book));

bookShelf = malloc(sizeof(Shelf) * 11);

bookShelf[3] = malloc(sizeof(Shelf));

bookShelf[3].books[7]->author = malloc(sizeof(char) * 50);

bookShelf[3].books[7]->author = malloc(sizeof(char));
```

```
Question 6
```

# Consider the following code:

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

int a = 32;
int b;
```

```
int func(int arg) {
    static int tmp = 0;
    tmp++;
    return tmp;
}

int main(int argc, char *argv[]) {
    int *p = malloc(sizeof(int));
    *p = 43;
    char *str = "where am I?";
    printf("%s\n", str);
    return 0;
}
```

Where are p and \*p stored in the program's virtual address space?

# Correct!

	Stack,	Heap
--	--------	------

- Stack, Stack
- Heap, Heap
- Heap, Stack
- Ode, Heap

Question 7

Below is a basic implementation of the Linux command "cat". This command is used to print the contents of a file on the console/terminal window.

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

int main(int argc, char* argv[]) {

    FILE *fp;
    if(2 != argc) {
        printf("Usage: cat <filename>\n");
        exit(1);
    }

    if ((fp = fopen(argv[1], "r")) == NULL) {
        fprintf(stderr, "Can't open input file %s\n", argv[1]);
        exit(1);
    }
    char buffer[256];
```

```
while (fgets(\underline{X}, 256, fp) != NULL)

fprintf(\underline{Y}, "%s", buffer);

fclose(\underline{Z});

return 0;
}
```

Which one of the following replacements for  $\underline{X}$ ,  $\underline{Y}$  and  $\underline{Z}$  will result in correct execution?

# Correct!

- X = buffer, Y = stdout, Z = fp
- $\bigcirc$  X = buffer, Y = fp, Z = stdout
- $\bigcirc$  X = buffer, Y = stdin, Z = fp
- $\bigcirc$  X = fp, Y = stdout, Z = buffer
- $\bigcirc$  X = buffer, Y = fp, Z = fp

# Question 8

1 / 1 pts

A process's memory segments in the virtual address space for a Linux based IA-32 system listed from high to low address are?

## Correct!

- Stack, Heap, Data, Code
- Stack, Data, Code, Heap
- Data, Stack, Heap, Code
- Heap, Stack, Code, Data
- Ode, Data, Heap, Stack

Quiz Score: 8 out of 8