Who wants to be a C programmer

Preparation for midterm

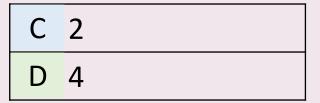
Game 2

Question 1. 500 points

```
int main() {
    int contestants[] = {1, 2, 3, 4};
    int *choice = contestants;
    contestants[0] = 2;
    contestants[1] = contestants[2];
    contestants[2] = *choice;
    printf("I'm going to pick contestant number %i\n", contestants[2]);
    return 0;
}
```

What contestant was picked by our automated system?

| Α | 3 |
|---|---|
| В | 1 |



Question 1. 500 points

```
int main() {
    int contestants[] = {1, 2, 3, 4};
    int *choice = contestants;
    contestants[0] = 2;
    contestants[1] = contestants[2];
    contestants[2] = *choice;
    printf("I'm going to pick contestant number %i\n", contestants[2]);
    return 0;
}
```

What contestant was picked by our automated system?

 A 3
 C 2

 B 1
 D 4

In myfile yourfile

What is the result of executing the command above?

- A Copy of *myfile* is created and is called *yourfile*
- B myfile is renamed to yourfile

- C The same file *myfile* now can be accessed though a different file name
- D A new file which combines content of both files is created

In myfile yourfile

What is the result of executing the command above?

- A Copy of *myfile* is created and is called *yourfile*
- B *myfile* is renamed to *yourfile*

- C The same file *myfile* now can be accessed though a different file name
- D A new file which combines content of both files is created

Question 3. 2,000 points

Variable *int* **p* is declared on the stack. What memory segment it points to when the program reaches the place indicated by a star?

| Α | Неар | |
|---|-------|--|
| В | Stack | |

| С | Constants |
|---|-----------|
| D | Globals |

Question 3. 2,000 points

Variable *int* **p* is declared on the stack. What memory segment it points to when the program reaches the place indicated by a star?

| Α | Heap | |
|---|-------|--|
| В | Stack | |

| С | Constants |
|---|-----------|
| D | Globals |

Question 4. 3,000 points

```
int x = 10;
void func() {
    int a = 0;
    int *p = &a;
    p = &x;
    p = malloc (5* sizeof (int) );
}
```

Variable *int* **p* is declared on the stack. What memory segment it points to when the program reaches the place indicated by a star?

| Α | Globals |
|---|---------|
| В | Stack |

| С | Неар |
|---|-----------|
| D | Constants |

Question 4. 3,000 points

```
int x = 10;
void func() {
    int a = 0;
    int *p = &a;
    p = &x;
    p = malloc (5* sizeof (int) );
}
```

Variable *int* **p* is declared on the stack. What memory segment it points to when the program reaches the place indicated by a star?

| Α | Globals | |
|---|---------|--|
| В | Stack | |

| С | Неар |
|---|-----------|
| D | Constants |

Question 5. 5,000 points

Variable *int *p* is declared on the stack. What memory segment it points to when the program reaches the place indicated by a star?

A Stack
B Heap

C Constants
D Globals

Question 5. 5,000 points

Variable *int* **p* is declared on the stack. What memory segment it points to when the program reaches the place indicated by a star?

A Stack C
B Heap D

C Constants
D Globals

Checkpoint 1 reached!

You have 5,000 points

```
int *fun (char *p) {
    int *int_ptr = malloc (sizeof (int) );
    if (p [0] == 'Z') {
        *int_ptr = 1;
    } else {
        *int_ptr = 5;}
    return int_ptr;
}
```

```
int main() {
    char letters[10] = "abc";
    int *x;
    x = fun (letters);
    return 0;
}
```

How much new memory is allocated when program executes line , and where is it allocated?

- A sizeof (int) on the heap
- B sizeof (int) on the stack

- C sizeof (int *) on the stack and sizeof (int) on the heap
- D 2*sizeof (int) on the heap

```
int *fun (char *p) {
    int *int_ptr = malloc (sizeof (int) );
    if (p [0] == 'Z') {
        *int_ptr = 1;
    } else {
        *int_ptr = 5;}
    return int_ptr;
}
```

```
int main() {
    char letters[10] = "abc";
    int *x;
    x = fun (letters);
    return 0;
}
```

How much new memory is allocated when program executes line , and where is it allocated?

A sizeof (int) on the heap

B sizeof (int) on the stack

C sizeof (int *) on the stack and sizeof (int) on the heap

D 2*sizeof (int) on the heap

Question 7. 10,000 points

```
int *fun (char *p) {
    int *int_ptr = malloc (sizeof (int) );
    if (p [0] == 'Z') {
        *int_ptr = 1;
    } else {
        *int_ptr = 5;}
    return int_ptr;
}
```

```
int main() {
    char letters[10] = "abc";
    int *x;
    x = fun (letters);
    return 0;
}
```

How much new memory is allocated when program executes line +, and where is it allocated?

A 10*sizeof (char) on the stack and 4*sizeof(char) in constants

B 10*sizeof (char) on the stack

C sizeof (char *) and 10*sizeof (char) on the stack

D 10*sizeof (char) in Constants

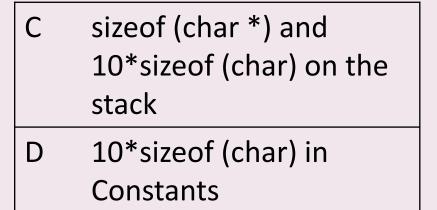
Question 7. 10,000 points

```
int *fun (char *p) {
    int *int_ptr = malloc (sizeof (int) );
    if (p [0] == 'Z') {
        *int_ptr = 1;
    } else {
        *int_ptr = 5;}
    return int_ptr;
}
```

```
int main() {
    char letters[10] = "abc";
    int *x;
    x = fun (letters);
    return 0;
}
```

How much new memory is allocated when program executes line , and where is it allocated?

| А | 10*sizeof (char) on the stack and 4*sizeof(char) in constants |
|---|---|
| | III COIIStairtS |
| В | 10*sizeof (char) on the |
| | stack |



Question 8. 15,000 points

```
int depth = 4;
int *p3 = &depth;
int x;
x = *p3 + 5;

depth = 1;
```

What are the values stored in variables depth, x, and *p3 at the moment of execution indicated by a star?

A depth: 4, x: 4, *p3: 4

B depth: 4, x: 9, *p3: 4

C depth: 9, x: 9, *p3: 9

D depth: 4, x: 9, *p3: 9

Question 8. 15,000 points

```
int depth = 4;

int *p3 = &depth;

int x;

x = *p3 + 5;

\Leftrightarrow

depth = 1;
```

What are the values stored in variables *depth*, *x*, and **p3* at the moment of execution indicated by a star?

| Α | depth: 4, x: 4, *p3: 4 |
|---|------------------------|
| В | depth: 4. x: 9. *p3: 4 |

D depth: 4, x: 9, *p3: 9

Question 9. 25,000 points

```
int depth = 4;
int *p3 = &depth;
int x;
int x;
x = *p3 + 5;
depth = 1;

♣
```

What are the values stored in variables depth, x, and *p3 at the moment of execution indicated by a star?

д depth: 1, x: 9, *p3: 4

B depth: 1, x: 9, *p3: 1

c depth: 1, x: 1, *p3: 4

depth: 1, x: 1, *p3: 1

```
int depth = 4;
int *p3 = &depth;
int x;
int x;
x = *p3 + 5;
depth = 1;
```

What are the values stored in variables depth, x, and *p3 at the moment of execution indicated by a star?

A depth: 1, x: 9, *p3: 4

B depth: 1, x: 9, *p3: 1

C depth: 1, x: 1, *p3: 4

D depth: 1, x: 1, *p3: 1

Question 10. 50,000 points

```
char \ c[6] = "ABCDE";
char \ *p = c;
char \ *s = p + 2;
printf("\%c\n", p[0]);
printf("\%s\n", p + 1);
printf("\%c\n", s[0]);
```

What output is printed to stdout?

A A
B
C
B A
BCDE
B

C The code will not run because of a compilation error

D A
BCDE
C

Question 10. 50,000 points

```
char c[6] = "ABCDE";

char *p = c;

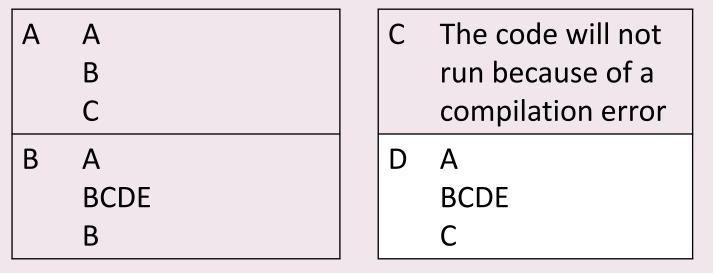
char *s = p + 2;

printf("%c\n", p[0]);

printf("%s\n", p + 1);

printf("%c\n", s[0]);
```

What output is printed to stdout?



Checkpoint 2 reached!

You have 50,000 points

```
char * p;

char a1 [] = "first";

char a2 [] = "second";

p = a1;

p [5] = '\0';
```

- A The code will not compile
- B The code will compile but always produce a run-time error
- C The code will compile and run with no errors
- D The code will compile and run, but sometimes give a run-time error

```
char * p;
char a1 [] = "first";
char a2 [] = "second";
p = a1;
p [5] = '\0';
```

- A The code will not compile
- B The code will compile but always produce a run-time error
- C The code will compile and run with no errors
- D The code will compile and run, but sometimes give a run-time error

```
char * p;
char a1 [] = "first";
char a2 [] = "second";
p = a1;
a2 = p;
```

- A The code will compile but always produce a run-time error
- B The code will not compile

- C The code will compile and run with no errors
- The code will compile and run, but sometimes give a run-time error

```
char * p;
char a1 [] = "first";
char a2 [] = "second";
p = a1;
a2 = p;
```

- A The code will compile but always produce a run-time error
- B The code will not compile

- C The code will compile and run with no errors
- D The code will compile and run, but sometimes give a run-time error

Question 13. 250,000 points

```
$ mkdir -
$ expr 9 * 4
```

What is printed when we execute these two shell commands?

A 36

B The output depends on what other files are in the current directory

C 5

D Error: non-integer argument

Question 13. 250,000 points

\$ mkdir -\$ expr 9 * 4

What is printed when we execute these two shell commands?

| Α | 36 |
|---|---|
| В | The output depends on what other files are in the current directory |

C 5

D Error: non-integer argument

```
int * ip;
long * lp;
double *dp;
printf ("%ld\n",sizeof (ip));
printf ("%ld\n",sizeof (lp));
printf ("%ld\n",sizeof (dp));
```

What is printed on the 32-bit system?

| Α | 4 |
|---|---|
| | 8 |
| | 8 |
| В | 4 |
| | 8 |
| | 4 |

| С | 4 | |
|---|---|--|
| | 4 | |
| | 4 | |
| D | 8 | |
| | 8 | |
| | 8 | |

```
int * ip;
long * lp;
double *dp;
printf ("%ld\n",sizeof (ip));
printf ("%ld\n",sizeof (lp));
printf ("%ld\n",sizeof (dp));
```

What is printed on the 32-bit system?

| Α | 4 | C 4 |
|---|---|-----|
| | 8 | 4 |
| | 8 | 4 |
| В | 4 | D 8 |
| | 8 | 8 |
| | | |

Question 15. One million points!

```
int double1 (int x) {
    x = x + x;
    return x;
}
int double2 (int *x) {
    *x = (*x) + (*x);
    return *x;
}
```

```
int main() {
    int i = 10;
    int j = 0;
    j = double1(i);
    printf ("i = %d, j = %d\n", i, j);
    i = 10;
    j = double2(&i);
    printf ("i = %d, j = %d\n", i, j);
    return 0;
}
```

What is the output of this program?

| Α | i = 10, j = 20 i = 10, j = 20 | С | i = 10, j = 20 i = 20, j = 20 |
|---|----------------------------------|---|----------------------------------|
| В | i = 10, j = 10 i = 20, j = 20 | D | i = 20, j = 20 i = 20, j = 20 |

Question 15. One million points!

```
int double1 (int x) {
    x = x + x;
    return x;
}
int double2 (int *x) {
    *x = (*x) + (*x);
    return *x;
}
```

```
int main() {
    int i = 10;
    int j = 0;
    j = double1(i);
    printf ("i = %d, j = %d\n", i, j);
    i = 10;
    j = double2(&i);
    printf ("i = %d, j = %d\n", i, j);
    return 0;
}
```

What is the output of this program?

| i = 10, j = 20 i = 10, j = 20 | | i = 10, j = 20 i = 20, j = 20 |
|----------------------------------|--|----------------------------------|
| i = 10, j = 10 i = 20, j = 20 | | i = 20, j = 20 i = 20, j = 20 |

Well done!

You are ready for the midterm