

# Who wants to be a C programmer

Preparation for midterm

Game 3

Question 1. 500 points

```
echo 6 + 3
```

What is printed when we execute this command?

A	9
B	6

C	6 + 3
D	Some sort of error message

Question 1. 500 points

```
echo 6 + 3
```

What is printed when we execute this command?

A	9
B	6

C	6 + 3
D	Some sort of error message

**The correct answer is C.**

Question 2. 1,000 points

```
first=Santa  
last=Clause  
name=first+last  
echo name is $name  
echo 'last is $last'
```

What is printed to stdout when we execute this shell script?

**A** name is Santa Clause  
last is Clause

**B** name is Santa Clause  
last is \$last

**C** name is first+last  
last is \$last

**D** name is first+last  
last is Clause

Question 2. 1,000 points

```
first=Santa  
last=Clause  
name=first+last  
echo name is $name  
echo 'last is $last'
```

What is printed to stdout when we execute this shell script?

A name is Santa Clause  
last is Clause

B name is Santa Clause  
last is \$last

C name is first+last  
last is \$last

D name is first+last  
last is Clause

**The correct answer is C.**

Question 3. 2,000 points

```
x=mon
x="$x day"
echo $x
y=thurs
y='$y day'
echo $y
```

What is printed to stdout when we execute this shell script?

A \$x day  
thurs day

B mon day  
\$y day

C \$x day  
\$y day

D mon day  
thurs day

Question 3. 2,000 points

```
x=mon
x="$x day"
echo $x
y=thurs
y='$y day'
echo $y
```

What is printed to stdout when we execute this shell script?

A \$x day  
thurs day

B mon day  
\$y day

C \$x day  
\$y day

D mon day  
thurs day

**The correct answer is B.**

Question 4. 3,000 points

```
if foo
then
    echo foo successful >&2
    bar
else
    echo sorry, foo failed >&2
    exit 1
fi
echo goodbye >&2
```

If *foo* program completed with code 1, what is printed to *stderr*?

A	sorry, foo failed
B	foo successful goodbye

C	sorry, foo failed goodbye
D	foo successful



Question 4. 3,000 points

```
if foo
then
    echo foo successful >&2
    bar
else
    echo sorry, foo failed >&2
    exit 1
fi
echo goodbye >&2
```

If *foo* program completed with code 1, what is printed to *stderr*?

A	sorry, foo failed
B	foo successful goodbye

C	sorry, foo failed goodbye
D	foo successful

**The correct answer is A.**

Question 5. 5,000 points

```
int *get_array5 () {  
    int a[5];  
    int i=0;  
    while (i++ < 5)  
        a[i] = i+1;  
    return (a);  
}
```

```
int main () {  
    int i=0;  
    int * p = get_array5 ();  
    while (i++ < 5)  
        printf ("%d ", *(p+i));  
}
```

What is printed when we run this code?

A 0 1 2 3 4

B An unpredictable sequence of numbers and sometimes seg fault

C 1 2 3 4 5

D Nothing will be printed: always run-time error

Question 5. 5,000 points

```
int *get_array5 () {  
    int a[5];  
    int i=0;  
    while (i++ < 5)  
        a[i] = i+1;  
    return (a);  
}
```

```
int main () {  
    int i=0;  
    int * p = get_array5 ();  
    while (i++ < 5)  
        printf ("%d ", *(p+i));  
}
```

What is printed when we run this code?

A 0 1 2 3 4

B An unpredictable sequence of numbers and sometimes seg fault

C 1 2 3 4 5

D Nothing will be printed: always run-time error

**The correct answer is B.**

# Checkpoint 1 reached!

You have 5,000 points

Question 6. 7,500 points

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

int main () {
    char course [6]="CSC209";
    int len;
    course [6] = 'H';
    len = strlen (course);
    printf ("%d\n",len);
    return 0;
}
```

What is printed when we run this code?

A 6

B 7

C Unpredictable number or some sort of run time error message

D Unpredictable number

Question 6. 7,500 points

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

int main () {
    char course [6]="CSC209";
    int len;
    course [6] = 'H';
    len = strlen (course);
    printf ("%d\n",len);
    return 0;
}
```

What is printed when we run this code?

A	6
B	7

C	Unpredictable number or some sort of run time error message
D	Unpredictable number

**The correct answer is C.**

Question 7. 10,000 points

```
void func() {  
    char *name = "ann";  
    ★  
    name = malloc(10);  
}
```

Variable *name* is declared on the stack. What memory segment it points to in the place indicated by a star?

A	Stack
B	Constants

C	Heap
D	Global

Question 7. 10,000 points

```
void func() {  
    char *name = "ann";  
    ★  
    name = malloc(10);  
}
```

Variable *name* is declared on the stack. What memory segment it points to in the place indicated by a star?

A	Stack
B	Constants

C	Heap
D	Global

**The correct answer is B.**



Question 8. 15,000 points


```
void func() {  
    char *name = "ann ";  
    name = malloc(10);  
    *  
}
```

Variable *name* is declared on the stack. What memory segment it points to in the place indicated by a star?

A	Stack
B	Heap

C	Constants
D	Globals

Question 8. 15,000 points

```
void func() {  
    char *name = "ann ";  
    name = malloc(10);  
      
}
```

Variable *name* is declared on the stack. What memory segment it points to in the place indicated by a star?

A	Stack
B	Heap

C	Constants
D	Globals

**The correct answer is B.**

Question 9. 25,000 points

```
void func() {  
    char name [] = "bob";  
    ★  
}
```

Variable *name* is declared on the stack. What memory segment it points to in the place indicated by a star?

A Heap

B Stack

C Constants

D Globals

Question 9. 25,000 points

```
void func() {  
    char name [] = "bob";  
    ★  
}
```

Variable *name* is declared on the stack. What memory segment it points to in the place indicated by a star?

A	Heap
B	Stack

C	Constants
D	Globals

**The correct answer is B.**

Question 10. 50,000 points

```
void helper (int *arr) {  
    arr = malloc(sizeof(int));  
}
```

Where is the variable `arr` stored and to what memory segment it is pointing to?

**A** arr is stored on the heap,  
pointing to the stack

**B** arr is stored on the heap,  
pointing to the heap

**C** arr is stored in globals,  
pointing to the heap

**D** arr is stored on the stack,  
pointing to the heap

Question 10. 50,000 points

```
void helper (int *arr) {  
    arr = malloc(sizeof(int));  
}
```

Where is the variable `arr` stored and to what memory segment it is pointing to?

A `arr` is stored on the heap, pointing to the stack

B `arr` is stored on the heap, pointing to the heap

C `arr` is stored in globals, pointing to the heap

D `arr` is stored on the stack, pointing to the heap

**The correct answer is D.**

# Checkpoint 2 reached!

You have 50,000 points

Question 11. 75,000 points

```
char fullname[30] = "Frederick";  
fullname[4] = '\0';  
printf ("%s\n", fullname);
```

What is printed?

A Frederick

B Nothing will be printed because of a run-time error

C Fred

D Fre



Question 11. 75,000 points

```
char fullname[30] = "Frederick";  
fullname[4] = '\0';  
printf ("%s\n", fullname);
```

What is printed?

A Frederick

B Nothing will be printed because of a run-time error

C Fred

D Fre

**The correct answer is C.**

Question 12. 150,000 points

```
int j;  
int x[5] = {4,2,7,8,9};  
for (j = 0; j <= 5; j++) {  
    fprintf ("%d\n", x[j]);  
}
```

What happens if we write this code?

A The code will not compile

B The code will compile and run, and have an unpredictable behavior

C The code will compile and run with no errors

D The code will compile but always produce a run-time error

Question 12. 150,000 points

```
int j;  
int x[5] = {4,2,7,8,9};  
for (j = 0; j <= 5; j++) {  
    fprintf ("%d\n", x[j]);  
}
```

What happens if we write this code?

A The code will not compile

B The code will compile and run, and have an unpredictable behavior

C The code will compile and run with no errors

D The code will compile but always produce a run-time error

**The correct answer is B.**

Question 13. 250,000 points

```
char *name = "Daniel";  
printf("%lu\n", sizeof(name));
```

If the size of char on my 32-bit machine is 1 byte, what is printed?

A	6
B	4

C	1
D	7

Question 13. 250,000 points

```
char *name = "Daniel";  
printf("%lu\n", sizeof(name));
```

If the size of char on my 32-bit machine is 1 byte, what is printed?

A	6
B	4

C	1
D	7

**The correct answer is B.**

Question 14. 500,000 points

```
int x, y;  
int *px, *py;
```

```
px = &x;  
py = &y;
```

A  $p = px + py;$

B  $p = px * py;$

C  $py = px + 2;$

D  $p = px + 10.0;$

Which of pointer arithmetic operations is legal?

Question 14. 500,000 points

```
int x, y;  
int *px, *py;
```

```
px = &x;  
py = &y;
```

- A  $p = px + py;$
- B  $p = px * py;$
- C  $py = px + 2;$
- D  $p = px + 10.0;$

Which of pointer arithmetic operations is legal?

**The correct answer is C.**

Question 15. **One million points!**

```
int mystery (int *a, int n) {  
    int result=0;  
    for ( ; n>0 ; n--, a++)  
        if (*a > result)  
            result = *a;  
    return result;  
}
```

```
int main () {  
    int arr [] = {1,2,3,4};  
    int x = mystery (arr, 4);  
    return 0;  
}
```

**A** It computes the maximum value in an integer array without modifying an array

**B** It computes the maximum value in an integer array, and modifies an array to point to the last array element

**C** It computes the maximum value in an integer array without modifying an array, but only if an array has at least one positive element

**D** The program will cause the run-time error, because the function tries to change the memory address of the array



Question 15. **One million points!**

```
int mystery (int *a, int n) {  
    int result=0;  
    for ( ; n>0 ; n--, a++)  
        if (*a > result)  
            result = *a;  
    return result;  
}
```

```
int main () {  
    int arr [] = {1,2,3,4};  
    int x = mystery (arr, 4);  
    return 0;  
}
```

A It computes the maximum value in an integer array without modifying an array

B It computes the maximum value in an integer array, and modifies an array to point to the last array element

C It computes the maximum value in an integer array without modifying an array, but only if an array has at least one positive element

D The program will cause the run-time error, because the function tries to change the memory address of the array

**The correct answer is C.**

# Well done!

You are ready for the midterm