

1. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
    int a[4] = {13, 14, 15, 19};
    printf("%d",a[1]);
    return 0;
}
```

(a) 0    (b) 4    (c) 14    (d) 13    (e) error

2. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
    int a[5] = {13, 15, 16, 17};
    printf("%d",a[4]);
    return 1;
}
```

(a) error    (b) 0    (c) 1    (d) 17    (e) 15

3. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
    int a[6] = {13, 15, 17, 19, 20, 21};
    printf("%d",a[6]);
    return 0;
}
```

(a) 0    (b) none of these    (c) 20    (d) 21    (e) error

4. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
    int a[6] = {1};
    printf("%d",a[3]);
    return 0;
}
```

(a) 0    (b) 1    (c) 3    (d) 6    (e) error

For the next **four** questions, choose the letter that shows the value that would be in x. Assume x and y are declared as integers. Choose “error” if you think a fragment will not compile and run. The fragments are unrelated (each question is separate from the others).

5. (2 pts)

```
x = -4 || 0;
```

- (a) 0    (b) -4    (c) -1    (d) error    (e) none of the above

6. (2 pts)

```
x = 5 % 3;
```

- (a) 2    (b) 1    (c) 3    (d) error    (e) none of the above

7. (2 pts)

```
x = 2 % 3;
```

- (a) 1    (b) 0    (c) 3    (d) 2    (e) none of the above

8. (2 pts)

```
y = 30 != 3;  
x = -3 && !y;
```

- (a) 1    (b) 0    (c) 27    (d) -3    (e) error

9. (2 pts) What is the last index of w that is initialized after the declaration shown?

```
char w[8] = "arbutus";
```

- (a) 7  
(b) 8  
(c) 9  
(d) none of the above  
(e) error

10. (2 pts) What is the output?

```
#include <stdio.h>  
int main(void)  
{  
    char a[40][10] = {"albatross", "spam", "nugent", "that"};  
    printf("%c", a[2][3]);  
    return 0;  
}
```

- (a) g  
(b) m  
(c) e  
(d) none of the above  
(e) error

11. (2 pts) Which of the following is a correct way to change c *after* the code shown?

```
int main(){  
    char c;  
    ...
```

- (a) `char c = "s";`
- (b) `c = 's';`
- (c) `c = strcpy(c, "s");`
- (d) `strcpy(c, "s");`
- (e) none of the above

12. (2 pts) Which of the following assigns a random value to x?

- (a) `random(x);`
- (b) `srand(x)`
- (c) `rand(x);`
- (d) `x=rand();`
- (e) none of these

13. (2 pts) Assume x contains a random number. Which of the following puts x in the range 0-100, inclusive?

- (a) `x %= 101;`
- (b) `x = x % 100;`
- (c) `x.range(100);`
- (d) `srand(&x, 200, 300);`
- (e) none of these

14. (2 pts) Assume x contains a random number. Which of the following puts x in the range 200-300, inclusive?

- (a) `x = x % (300 - 200) + 200;`
- (b) `x = 200 + x % 101;`
- (c) `x = x % 100 + 200;`
- (d) `srand(&x, 200, 300);`
- (e) none of these

15. (2 pts) How many numbers will binary search examine to find one member of a sorted list of 100,000 numbers?

- (a) approximately 6
- (b) approximately 16
- (c) approximately 17
- (d) approximately 21
- (e) none of the above

16. (2 pts) How many numbers will binary search examine to find one member of a sorted list of 10,000,000 numbers?

- (a) approximately 14
- (b) approximately 16
- (c) approximately 17
- (d) approximately 18
- (e) none of the above

17. (2 pts) How many numbers will binary search examine to find one member of a sorted list of 1,000,000,000 numbers?

- (a) approximately 21
- (b) approximately 22
- (c) approximately 24
- (d) approximately 30
- (e) none of the above

## Short Answers

Consider the **partial** program below. Imagine that you are going to add some functions to it. Make up function names, and use parameters from the existing code.

```
#include <stdio.h>

/* Terry Harvey CISC105-20 TA: Liric Waterchard*/

int main(){

    int someData[100] = {1,2,3,4,5,6,7};
    int one = 5;
    ...
}
```

18. (4 pts) Write a prototype for a function that can change any element in someData. Do not write the definition.
19. (4 pts) Show how you could call the function from 18 in main().
20. (4 pts) The function printInteger takes a single integer as argument. Show a call to printInteger that will print the 4 from someData.
21. (4 pts) A program is running and has 200Kb of memory available for an array. What is the maximum number of integers an array could hold and still fit? Show your work.
22. (6 pts) The file **data.txt** is in your home directory. Show the complete shell commands, in order, that would get you to your home directory, move the file to the 105 directory of your webpage and then make it readable.

MC answers: CBEAEADBACBDABCE

23. (20 pts) Fill in the blanks. Be careful that your parameter names match the names used in the body of the function definition.

```
/*
 * Recursive function for searching an integer array sorted
 * small (left) to large (right).
 * Returns -1 if key is not found.
 */
int binarySearch(_____
{

    int mid = _____

    if (_____) return -1;

    if (data[mid] < key)

        return _____

    else if (data[mid] > key)

        return _____
    else return mid;
}
```

## Strings and char arrays in C

(20 pts) Read the following program description carefully. Then fill in the code framework on the next page. Do not change the framework if you want full credit.

Copy two words from one array, **data**, into two separate arrays, ONE character at a time. Print the two words with only one space between them.

- The two words in data, and the number of spaces, may change.
- There will not be any spaces before the first word or after the second word.
- The FIRST word will only contain alphabetical characters.
- Assume the words will correctly fit in word1 and word2.
- Assume all whitespace is the space character.
- Assume all letters are lower case.

Example output:

```
% ./a.out
The words are: first more.14$
%
```

**Scratch area (work it out here, code on next page)**

```

#include <stdio.h>

/* See comments on previous page */

int main(){

    char word1[10]; /* These will be C strings. */
    char word2[10];
    int i = 0, j = 0, k = 0; /* You may not need all three vars */

    char data[50] = "first    more.14$";

    /* Put the first word from data into word1 */


    /* Get past the spaces (how will you know when you are past?) */


    /* Put the second word into word2. The second word may contain
       non-alphabetic characters. */


    return 0;
}

```