

Name\_\_\_\_\_Section\_\_\_\_\_

## General Instructions

- DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
- Turn off any noise making device, especially **CELL PHONES**. You may lose up to one letter grade if your device disturbs the peace of the exam.
- You have 50 minutes. **Pace yourself**, and pay attention to the point values.
- The exam has 25 multiple choice questions for 51% , and 49% programming.
- Do problems you are confident about first. If you finish the problems you know, write what you do know about other problems to gain partial credit; but erroneous information may detract from that credit or irritate the grader, so don't make stuff up.
- Read *all* the directions *carefully* on each problem.
- Often writing a fast, rough version of a program in English or pseudocode will make your C coding faster and more accurate. It also enables me to give partial credit in some circumstances.
- You may assume that input will not produce errors for the procedures described, unless the questions say otherwise.
- Do not do unnecessary testing. For example, testing for both  $x < 0$  and  $x \geq 0$  instead of using one test and then `else` would be considered unnecessary testing.
- Carpe A!

## What is the output? (Multiple choice)

Each of the questions in this section consists of a short C program.

For each one, either select the answer that matches the output of the program, **or**, if the program contains an error that would prevent you from determining the output, select “error” as your answer.

1. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
    char a[40];
    printf("%c",a[9]);
    return 0;
}
```

- (a) \0    (b) 0    (c) ^@    (d) !#[    (e) error

2. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
    int b[6] = {13, 15, 17, 19};
    printf("%d",b[4]);
    return 0;
}
```

- (a) 0    (b) 19    (c) 21    (d) '\0'    (e) error

3. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
    int data[5] = {3, 1, 5, 9};
    printf("%d",data[5]);
    return 1;
}
```

- (a) -1    (b) 0    (c) 1    (d) 9    (e) error

4. (2 pts) What is the last index of w that is initialized after the declaration shown?

```
char w[10] = "abacus";
```

- (a) 6    (b) 7    (c) 8  
(d) 9    (d) 10

5. (2 pts) If the variable x is of type int, which of the following expressions would print the address of that variable?

- (a) printf("%d", x);    (b) printf("%c", &x);    (c) printf("%a", &x);  
(d) printf("%p", &x);    (e) none of the above

6. (2 pts) pts. Suppose you have the declarations below. You are told that the address w1 is AB00. As discussed in class, which of the following is a likely address of w3?

```
char w1[] = "eggover";  
char w2[] = "e t c .";  
char w3[] = "declassse";
```

- (a) AB08  
(b) ABg  
(c) AC00  
(d) AB10  
(e) none of the above

7. (2 pts) pts. Suppose you have the declarations below. How many bytes are required to hold a?

```
int a[10];  
int b[8];  
int c[21];
```

- (a) 10  
(b) 40  
(c) 160  
(d) none, a is on the heap  
(e) none of the above

Consider the following declaration when answering the next two questions.

```
struct kitten {  
    int legs;  
    double weight;  
    char name[100];  
};
```

8. (2 pts) Which of the following is a correct declaration of a new data structure of the type shown above?
- (a) `struct kitten a;`
  - (b) `struct kitten b = {4, "SheWhoMustBeObeyed"};`
  - (c) `kitten c = {4, 17, "SheWhoMustBeObeyed"};`
  - (d) `struct c = {4, 17, "SheWhoMustBeObeyed"};`
  - (e) none of the above
9. (2 pts) Assume you have correctly made a new struct of the type above, named x. Which of the following correctly gives x a new weight?
- (a) `struct x.weight = 8.5;`
  - (b) `x.weight(8.5);`
  - (c) `x.weight = 6;`
  - (d) cannot re-assign weight
  - (e) none of the above
10. (2 pts) When a struct is passed as a parameter to a function using a function call, C uses
- (a) pass by name
  - (b) pass by assignment
  - (c) pass by reference
  - (d) pass by value
  - (e) none of the above
11. (2 pts) Which of the following function prototypes represents a function that can modify the contents of an integer array declared in main() as `int data[5];`?
- (a) `double f(int array[]);`
  - (b) `int f(int data);`
  - (c) `void f(struct int *data);`
  - (d) `int f(int *data[]);`
  - (e) none of the above

12. (2 pts) Which of the following function prototypes represents a function that can modify the contents of an integer declared in main() as `int input;` ?

- (a) `double f(int input);`
- (b) `int f(int data);`
- (c) `void f(const int data);`
- (d) `int f(int data, int size);`
- (e) none of the above

13. (2 pts) Selection sort works in a series of passes over an array. Choose the answer that shows how this array will appear after the **first** pass of selection sort (as demonstrated in class). Assume we are sorting small (left) to large (right).

original array: [5 2 8 4 1 3 7]

- (a) 1 5 2 8 4 3 7
- (b) 5 2 3 4 1 7 8
- (c) 5 8 4 3 7 1 2
- (d) 5 2 7 4 1 3 8
- (e) none of the above

14. (2 pts) Selection sort works in a series of passes over an array. Choose the answer that shows how this array will appear after the **third** pass of selection sort (as demonstrated in class). Assume we are sorting small (left) to large (right).

original array: [5 2 8 4 1 3 7]

- (a) 1 2 3 5 8 4 7
- (b) 1 2 3 4 5 7 8
- (c) 2 1 3 4 5 7 8
- (d) 5 8 7 1 2 3 4
- (e) none of the above

15. (3 pts) Consider the following program. Your task is to decide which set of three answers below goes in place of 111, 222, and 333 to make this program print the alphabet.

```
#include <stdio.h>
int main(){
    int i;

    for (i = 111; i <= 222; i++) /* replace 111, 222, and 333 */
        printf("%c", 333);
    return 0;
}
```

- (a) 'A', 'Z', and i
- (b) 'A', 'Z', and getchar(i)
- (c) 0, 25, and i
- (d) 1, 26, and i
- (e) none of the above

16. (2 pts) Consider the following fragment of a man page. What is the type of pntnark?

```
pntark(3)          TGH Library Functions Manual      pntark(3)

NAME
    pntark, pntnark - these functions paint arks

LIBRARY
    Standard C Library (libc, -lc)

SYNOPSIS
    #include <pnting.h>

    char *
    pntark(const char *s1, const char *s2);

    char *
    pntnark(const char *s1, const char *s2, size_t len);
```

- (a) int
  - (b) const
  - (c) char \*
  - (d) const char \*s1
  - (e) none of the above
17. (2 pts) Consider the above fragment of a man page. What is the name of the second parameter to pntark?
- (a) len
  - (b) const char \*s2
  - (c) char \*s2
  - (d) the second parameter has no name
  - (e) none of the above

## File input and output: 4 questions

Suppose you have an input file called `data.dat`. Each line in is supposed to contain an integer value. Your task is to read in all of these lines of input and find the sum. The questions must be read in sequence (but you may answer in any order you wish).

18. (2 pts) Read this question carefully: Which of the following *declares a variable* that could be used to access data currently stored in `data.dat`?
- (a) `FILE *input;`
  - (b) `FILE *data.dat;`
  - (c) `input = fopen("data.dat", "r");`
  - (d) `input = fopen("data.dat", "w");`
  - (e) none of the above
19. (2 pts) Which of the following opens the file `data.dat` so that the program can do operations such as `fscanf`?
- (a) `FILE *input;`
  - (b) `FILE *data.dat;`
  - (c) `input = fopen("data.dat", "r");`
  - (d) `fgets("data.dat", 80, input);`
  - (e) none of the above
20. (2 pts) Which of the following evaluates to true when there was a problem opening the file (e.g. `data.dat` doesn't exist)?
- (a) `(input != NULL)`
  - (b) `(input == NULL)`
  - (c) `(input == EOF)`
  - (d) `(input != EOF)`
  - (e) none of the above
21. (2 pts) Suppose a variable has been declared as `int x;`  
Which of the following reads an integer from the file into this variable?
- (a) `scanf("%d", x);`
  - (b) `scanf("%d", &x);`
  - (c) `fscanf(input, "%d", x);`
  - (d) `fgets(x, 40, stdin);`
  - (e) none of the above



Your task: if the code fragment has no errors, select the answer that corresponds to the value of x. If the code fragment won't compile or will produce a run-time error, choose the answer "error". The code fragments are unrelated.

22. (2 pts)

```
double x = 13 / 5;
```

(a) 3.0 (b) 2.6 (c) 1.0 (d) 2.0 (e) error

23. (2 pts)

```
double x = 9 % 6;
```

(a) 3.0 (b) 54.0 (c) 1.0 (d) 2.0 (e) error

24. (2 pts)

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

(a) 3 (b) 0 (c) 1 (d) 2 (e) error

25. (2 pts)

```
int x, y=0;  
for(x = 0; x < 100; x++)  
    y++;
```

(a) 99 (b) 100 (c) 101 (d) 200 (e) error

```

#include <stdio.h>
#define SIZE 7
/* This program demonstrates selection sort on an integer array. */

void selectionSort(int input[], int size);
int findMaxIndex(int input[], int size);

A)_____

int main(){
    int i;
    int data[SIZE] = {5,2,8,4,1,3,7};

    B)_____
    printArray(data, SIZE);
    return 0;
}

void selectionSort(int input[], int size){
    int i;
    int maxIndex, temp;
    for(i = size; i > 0; i--) {

        C)_____

        temp = input[maxIndex]; /*here is the swap*/

        D)_____

        E)_____
    }
    return;
}

int findMaxIndex(int input[], int size){
    int i;
    int maxIndex = 0;
    for ( i = 1; i < size; i++)
        if (input[i] > input[maxIndex])
            maxIndex = i;
    return maxIndex;
}

```

## **Selection Sort, 2 questions**

26. (20 pts) For the selection sort program on the opposite page, fill in the five blanks. Only fill in the blanks. Read the question below before you fill in the blanks.
27. (10 pts) Notice that a function you need (printArray) is missing. Recall that printArray prints an array of arbitrary size.

Write only the definition of the function on THIS PAGE. Write the rest of what you need in the blanks on the opposite page.

PUT THE DEFINITION HERE:

## String functions

Write the definition of a function **strEquals** that takes two strings of unknown size as parameters. It returns 1 if the strings are identical, and 0 if they are not (note this function is **not** like strcmp). Your function should not do unnecessary testing.

Hints: Traverse both strings using an index variable, checking for equality on the way. If you get to the end of one string before the other, they are not the same. Draw yourself a picture.

28. (19 pts)