

Part I: Multiple Choice (24 points)

Answer all of the following questions. READ EACH QUESTION CAREFULLY. Fill the correct bubble on your mark-sense sheet. Each correct question is worth 2 points. Choose the one BEST answer for each question. Assume that all given C code is syntactically correct unless a possibility to the contrary is suggested in the question.

Remember not to devote too much time to any single question, and good luck!

1. What is true about a `for` loop?

- A. The loop variable must be an integer.
- B. It is possible that the loop will never terminate.
- C. The `for` loop may never appear within a `while` loop.
- D. A `while` loop may never appear inside the `for` loop.
- E. `for` loops may not be nested.

2. Examine the following code fragment

```
int i;
int weird[4];
weird[0] = 3
for (i=1; i<4; i++){
    weird[i] = 2 * weird[i-1];
}
printf("%d %d\n", weird[1], weird[3]);
```

what does it output?

- A. 3 24
- B. 1 12
- C. 6 12
- D. 12 24
- E. None of the above

3. Which of the following will be the certain result of failing to fill in properly your name, student ID, section number, and exam version on your Scantron answer sheet?

- A. Your head TA will have the distinguished honor of eating your exam.
- B. You get a perfect score on your exam.
- C. Isaac will give you 5 dollars because of your stunning performance on the exam.
- D. Your fellow students will be amazed at your incredible performance on the exam.
- E. You will fail the exam.

4. What is true about functions in C

- A. THE ARGUMENTS ARE EVALUATED AND COPIED INTO THE PARAMETERS
- B. IF A PARAMETER IS CHANGED, ITS CORRESPONDING ARGUMENT IS CHANGED IN THE FUNCTION THAT CALLED IT
- C. FUNCTIONS DEFINITIONS CAN BE NESTED
- D. A AND C ARE CORRECT
- E. ALL OF THE ABOVE ARE CORRECT

5. Examine this following body of a function:

```
...{
    int *x;
    char y;
    y = *s;
    x = &n;
    if (y == 'A')
        return *x;
    return -n;
}
```

What would be a plausible prototype for this function?

- A. `int f(int *n, char s);`
- B. `void f(char *s);`
- C. `int f(double n, char s);`
- D. `int f(int n, char *s);`
- E. `void f(int *x, int n, char *s);`

6. Examine the following loops

i)

```
for (i=0; i<4; i++){
    for (j=0; j<=i; j++){
        printf("*");
    }
    printf("\n");
}
```

ii)

```
for (i=3; i>=0; i--){
    for (j=3; j>=i; j--){
        printf("*");
    }
    printf("\n");
}
```

These two loops

- A. print the same figure
- B. print figures that are mirror images of each other
- C. print completely unrelated figures
- D. print the same shape, but one is smaller than the other

7. Imagine a language named C- - which was exactly like C, but did not have a while loop. Which of the following is true about our fictional language?

- A. There are things we can do in C that we cannot do in C--.
- B. Everything we can do in C we can do in C--.
- C. Everything we can do in C we can do in C--, but if we also lost the do...while loop we would not.
- D. There are things you can do in C-- that you cannot do in C.
- E. My CSE 142 TA is named Ralph.

8. Examine the following outline of a function:

```
int wisconsin(int cheese, int *milk, double *moo){
    int bessie;
    int *daisy;

    /* some code here */
}
```

Which assignment would be legal to make inside this function?

- A. `bessie = moo;`
- B. `*bessie = milk;`
- C. `daisy = &cheese;`
- D. `bessie = &milk;`
- E. `&daisy = moo;`

9. Examine the following program:

```
int bacchus(int *m, int n, int *p){
    int *s = &n;
    *p = 2;
    p = s;
    *p = n + 1;
    return (*m) * (*s);
}

int main(void){
    int x = 3;
    int y;
    y = bacchus(&x, x, &x);
    printf("%d %d", x, y);
    return 0;
}
```

What does it output?

- A. 2 6
- B. 2 3
- C. 2 8
- D. 3 6
- E. 4 8

10. Given this function definition

```
int square(int x){
    x = x * x;
    return x;
}
```

and the following main function

```
int main(void){
    int x = 3;
    /* <--- find the square of 3 here */
    printf("3 squared is %d\n", x);
    return 0;
}
```

how should we fill in the missing line?

- A. square(x);
- B. x = square(x);
- C. square(3);
- D. *x = square(x);
- E. None of the above

11. Which of the following is not true about a function's parameters and arguments?

- A. It is possible to define a function with any number of parameters.
- B. The names of the parameters and the names of variables passed to the function as arguments must match.
- C. There must be the same number of arguments and parameters.
- D. The first argument matches with the first parameter, the second argument with the second parameter, etc.
- E. None of them (they're all true)

12. Examine the following code fragment:

```
int x, int *y;
...
scanf("%d", &x); /* I */
printf("%d", x); /* II */
scanf("%d", y); /* III */
printf("%d", *y); /* IV */
```

What lines are syntactically correct?

- A. I and II
- B. I, II, and IV
- C. I, III, and IV
- D. I, II, and III
- E. All of them

Part II: Programming Questions (26 points)

Write C code for the following two problems.

13. (13 points)

A frog falls down a well that is `numfeet` feet deep and is trying to get out. On the frog's first jump he leaps 1 foot, and on each successive jump he doubles his jump distance. Unfortunately, after making each jump he slips one foot back down the well. For example, his first 3 jumps will look like this:

Jump	Jump Distance	Distance Slipped	New Height
1	1	1	0
2	2	1	1
3	4	1	4

The frog gets out of the well once he makes it as high as the top. For example, if the well were 1 foot deep, then the frog would make it out on his first jump. (Remember, the frog may make it out of the well before he slips back!)

Write a function that takes `numfeet` as a parameter, and returns the number of jumps the frog makes before getting out of the well. Your function should do no I/O and not call any other functions *unless you write them!*

14. (13 points)

The State of Washington needs your help tallying votes! Write a program to help them determine the winner of an election. Your program should have 3 functions.

- The function `vote()` should do all of the user input and tally the votes. The user should enter 'd' for Democrat, 'r' for Republican. We won't worry about those other parties, since they never matter anyway. You should loop until the user enters 'q'. Once the votes are tallied, the totals should be communicated back to the calling function.
- The function `results()` should determine the percentage of the vote that went to Republicans and Democrats. Then these percentages should be printed for the user. Don't worry about declaring a winner—we'll assume that the user can determine that from the percentages.
- Finally, you should write a main function that properly calls your other two functions.

Scratch Page: