

Practice Test #1, COP3514

This practice test has more questions than a regular midterm. It has 15 short-answer questions and 6 free-response questions. You can expect 10 short-answer questions and 5 free-response questions in the midterm.

1. What is the value of sum after the following program fragment is executed?

```
int i, j, sum=0;
for(i=0; i<3; i++) {
    for(j=1; j<4; j++) {
        if((i+j)%2==1) {
            continue;
        }
        sum++;
    }
}
```

Answer:

2. What is the output of the following program fragment?

```
int i, j;
for(i=0; i<5; i++) {
    for(j=0; j<=i; j++) {
        if(j >= 3) {
            break;
        }
        printf("*");
    }
    printf("\n");
}
```

Answer:

3. What is the output of the following program fragment if the value of x is 12?

```
if(x <= 10)
    printf("%d\n", x);
else if(10 > x && x <= 20)
    printf("%d\n", x*2);
else
    printf("%d\n", x*3);
```

Answer:

4. Given the following code, what is the final value of sum?

```
int i, j, sum = 0;
for(i=0; i<3; i++) {
    for(j=0; j<3; j++) {
        if(j==1) continue;
        sum++;
    }
}
```

Answer:

5. What is the value of sum after the following program fragment is executed?

```
int i, k, sum=0;
for(i=1; i<3; i++) {
    for(k=0; k<3; k++) {
        sum += k;
        if(k%2 == 1)
            break;
    }
}
```

Answer:

6. Consider the following recursive function:

```
int mystery(int k) {
    if(k == 1)
        return 0;
    else
        return (1 + mystery(k/2));
}
```

What value is returned by the call mystery(16)?

Answer:

7. What is the output of the following code fragment?

```
int v1 = 2, v2 = -1;

int *p1;
```

```
int *p2;

p1 = &v1;
p2 = &v2;
p2 = p1;

printf("%d %d", *p1, *p2);
```

Answer:

8. What is the value of sum after the code below is executed?

```
int i, sum = 0;
for(i = 0; i < 6; i++) {
    switch (i) {
        case 0:
        case 1:
        case 3:
        case 5:
            sum++;
            break;
        case 2:
        case 4:
            sum += 2;
            break;
    }
}
```

Answer:

9. What is the output of pb(13)?

```
void pb(int n) {
    if(n != 0) {
        pb(n/2);
        printf("%d", n%2);
    }
}
```

Answer:

10. What is the output of the code segment?

```
i = 5;
while(--i > 0);
    printf("%d", i);
```

Answer:

11. What is output of the following program?

```
#include <stdio.h>

int i =2;

void f();

int main(void){
    int i = 5;
    i++;
    f();
    printf("%d\n", i);
    return 0;
}

void f(){
    i++;
    printf("%d\t", i);
}
```

Answer:

12. What is the output of the following program?

```
#include <stdio.h>

int i=9;

int main(){
    if(i >= 3) {
        int i=2;
        printf("%d ", i);
        i++;
        if(i >= 3) {
            int i=7;
            printf("%d ", i);
        }
        if(i >= 5) {
            i += 2;
        } else {
            i += 3;
        }
        printf("%d ", i);
    }
}
```

```

        printf("%d ", i);
        return 0;
}

```

Answer:

13. Which of the function definitions will return the index of the largest value of an integer array?

a)

```

int find_largest(int a[], int n) {
    int largest_index = 0;
    int i;
    for (i = 1; i < n; i++)
        if (a[i] > a[i+1])
            largest_index = i;
    return largest_index;
}

```

b)

```

int find_largest(int a[], int n) {
    int largest_index = 0;
    int i;
    for (i = 1; i < n; i++)
        if (a[i] > a[largest_index])
            largest_index = i;
    return largest_index;
}

```

c)

```

int find_largest(int a[], int n) {
    int largest_index = a[0];
    int i;
    for (i = 1; i < n; i++)
        if (a[i] > a[i+1])
            largest_index = a[i];
    return largest_index;
}

```

d)

```

int find_largest(int a[], int n) {
    int largest_index = 0;
    int i;
    for (i = 1; i < n; i++)
        if (a[i] > largest_index)
            largest_index = a[i];
}

```

```
    return largest_index;
}
```

Answer:

14. Which of the following defines a recursive function that prints numbers in separate lines from 1 to n in ascending order? Assume n is a positive number.

a)

```
int print_asc(int n) {
    if(n==1) {
        return 1;
    } else {
        return n;
        print_asc(n-1);
    }
}
```

b)

```
void print_asc(int n) {
    if(n==1) {
        printf("1\n");
    } else {
        printf("%d\n", n);
        print_asc(n-1);
    }
}
```

c)

```
void print_asc(int n) {
    if(n==1) {
        printf("1\n");
    } else {
        print_asc(n-1);
        printf("%d\n", n);
    }
}
```

d)

```
int print_asc(int n) {
    if(n==1) {
        return 1;
    } else {
        print_asc(n-1);
        return n;
    }
}
```

Answer:

15. Which of the following defines a recursive function for calculating the sum of number from 0 to n? Assume n is a non-negative number.

a)

```
int sum(int n) {  
    if(n==0)  
        return 0;  
    else  
        return sum(n+(n-1));  
}
```

b)

```
int sum(int n) {  
    if(n==0)  
        return 0;  
    else  
        return n+sum(n-1);  
}
```

c)

```
int sum(int n) {  
    if(n==0)  
        return 0;  
    else  
        return n+(n-1);  
}
```

Answer:

Free Response Questions

16. Point out the four compilation errors in the following program and explain why each one is not correct. Use the line numbers to indicate where the errors are.

```
1. #include <stdio.h>
2. #define 2PI 6.28318
3. double main() {
4. int i=4.5, *p;
5. double x;
6. p = *i;
7. x += a;
8. printf("%d %lf\n", *p, x);
9. return 0;
10. }
```

17. Write a recursive function that computes x^n , using the formula $x^n = x \times x^{n-1}$. Assume n is a non-negative number.

```
double power(double x, int n) {
```



```
}
```

18. Write a function that reverses the elements of an integer array **a**. For example, if array **a** contains elements {2, 3, 6, 8, 5}, the function will swap the elements of the array so it contains {5, 8, 6, 3, 2}. The function has the following prototype: **n** is the length of the array, **a** is the integer array. You are not allowed to use any other arrays except array **a** to solve this problem.

```
void reverse(int a[], int n) {
```

```
}
```

19. Write the definition for a function that will return the lower case of **ch** if **ch** is an upper case alphabetic letter, otherwise return the original character. Library functions are NOT allowed in this question.

```
char to_lower_case(char ch) {
```

```
}
```

- 20.** Complete the following function that compares elements of two integer arrays **a** and **b**, and stores the elements in array **c** that are in both **a** and **b**. For example, if array **a** contains elements {1, 2, 3}, and array **b** contains elements {3, 2, 6, 7}, array **c** should contain {2, 3}. Assume array **a** has **n1** distinct elements and array **b** has **n2** distinct elements. Array **c** has the length of **n1** or **n2**, whichever is smaller. The function returns the actual number of elements in array **c**.

```
int find_elements(int a[], int n1, int b[], int n2, int c[]) {
```

```
}
```

21. Fill the blanks in the function below so it sorts the elements of an integer array **a** in non-decreasing order using selection sort (at each iteration, find the smallest element in a subarray and move this element to the beginning of the subarray). For example, if **a** contains the elements {2, 3, 6, 3, 5}, the function will sort the elements of the array so it contains {2, 3, 3, 5, 6}. The function has the following parameters: **a** is the integer array, **n** is the length of **a**. You are not allowed to use any other arrays except array **a** to solve this problem.

```
void my_sort(int a[], int n) {  
  
    int i, j;  
    for(i=0; i < n-1; i++) {  
  
        int smallest = i;  
  
        // find the index of the smallest element in the subarray  
        for(j=_____ ; _____ ; _____) {  
            if(a[j] < a[smallest]) {  
                smallest = j;  
            }  
        }  
  
        // move smallest to the correct location  
  
  
  
  
  
  
    }  
}
```