Practice Test 3 Answer Keys

1) if(p->value == n) count++;
2) “The”
3) n = 2 n = 1 n = 0 n = -1 n = 0 n = 1
4) Line 14
5) int *a= malloc(n* sizeof(int));
6) 6
7) 5
8) list == NULL //or if(list == NULL)
9) double integrate(double (*f)(double x), double a, double b);

10) A
11) D
12) A
13) D
14) B
15) fscanf(fp, “%s %d %d %d\n”, name, &month, &day, &year);
16) A
17) A

18)

```
int count_characters(char *filename) {
    char str[1001];
    FILE* pFile;
    int count = 0;
    pFile = fopen(filename, "r");
    if (pFile == NULL) {
        printf ("Error opening file\n");
        return 0;
    }
     while ( fgets (str, 1000, pFile ) != NULL )  {
        count += strlen(str);
    }
   fclose(pFile);
    return count;
}
```

19)

```
void search(struct part inv[], int np) {
      int i, number;
      int found =0;
      printf("Enter part number: ");
      scanf("%d", &number);
      for(i = 0; i < np; i ++) {
            if (inv[i].number == number){
                  printf("%s", inv[i].name);
                  printf("%d", inv[i].on_hand;
                  found = 1;
             }
      }
      if(!found)
            printf("part not found");
}
```

20)

```
double derive(double (*f)(double), double x)
{
    double stepSize = 0.01;

    double x1 = x - stepSize;
    double x2 = x + stepSize;
    return  (f(x2)- f(x1))/ (2 *stepSize);
}
```

21)

```
struct node *move_last_to_first(struct node *list) {
      struct node* prev, *cur;

        if(list == NULL) return list;   //the list is empty

        for(prev = NULL, cur=list; cur->next !=NULL;
                prev = cur, cur = cur->next)
        ;
        if(prev == NULL)  //there is only one node in the list
                return list;
```

```
        else{
                prev ->next = NULL;
                cur ->next = list;
                return cur;
        }
}
```

22)

```
struct node *delete_all(struct node *list, int n) {
    int old_size;
    do {
        old_size = size(list);
        list = delete(list, n);
    } while(old_size != size(list));
    return list;
}
```

23)

```
#include <stdio.h>
int main(int argc, char *argv[]) {
  FILE *fp;
  fp = fopen(argv[1], "a");
  if(fp == NULL) {
      printf("Error opening file. ");
      return 1;
  }
  fprintf(fp, "That's all, folks!\n");
  fclose(fp);
  return 0;
}
```