CIS 1201 Mock Final Exam - May 22, 2021

Name:	ANSWER KEY
Group #:	

SCORES		
Test 1	70	
Test 2	70	
TOTAL	**140**	

TEST 1

1.
2.
3.
	scanf("%c",*c);
	printf("%c",**c);
	1

4. 5. 6-7. 8-9.	8 20 8 13
10. 11.	ctype.h stdio.h
12-14.	if (islower(*str)!=0) *str=toupper(*str);

15-16.
17-18.
19-20.
21.
	memcpy(Accounts2,Accounts1,10*sizeof(struct user);
	tempAcc1 = Accounts2[3];
	strcpy(tempAcc2->username,Accounts2[9].username);
	string.h

22. 23. 24-25. 26-27.	8084 123 25 48
28-30.	return number & 1 << n - 1;
31. 32. 33. 34. 35-36.	2 8 8 12 colorlist=(struct color*)malloc(sizeof(struct color));
37-38.	1000
39.	2
40-41. 42-43.	12 0.0, 2.0, 3.0, 6.0, 4.0, 9.0
44-45. 46-48.	7 9,2,9
49-50. 51. 52. 53. 54. 55. 56. 57. 58-59. 60-62.	124 15 True False -6 2FE030 2FE014 2FE018 int(*ptr)[5]=arr; -6
63. 64. 65. 66-67. 68. 69-70.	numList*A numList B numList* trav=*B;*trav!=NULL;trav=&(*trav)->nextnum *trav=*A *A=NULL

TEST 2 - Problem A | Obtained score:

```
/*For 15 points - use recursion*/
int computePoly(polynomial poly, int x){
   if (poly == NULL)
      return 0;
   int temp = 1, ctr;
   for(ctr = poly->expo; ctr > 0; ctr--){
      temp *= x;
   }
   temp *= poly->coe;
   return temp + computePoly(poly->ptr, x);
}


/*For 12 points - without recursion*/
int computePoly(polynomial poly, int x){
   polynomial trav;
   int sum = 0, temp, ctr;
   for(trav = poly; trav != NULL; trav = trav->next){
      temp = 1;
      for(ctr = poly->expo; ctr > 0; ctr--){
         temp *= x;
      }
      temp *= poly->coe;
      sum += temp;
   }
   return sum;
}
```

TEST 2 - Problem B | Obtained score:

```
int checkStrings(twostrings *A){
   int x, ctr = 0, countArr[26] = {0}; /*can use calloc(), but should not forget to free()*/
   for(x = 0; A->str1[x] != '\0'; x++){
       countArr[A->str1[x] - 'A']++;
       countArr[A->str2[x] - 'A']--;
   }
   for(x = 0; x < 26 ; x++){
       while(countArr[x] < 0){
          A->extras[ctr] = countArr[x] + 'A';
          countArr[x]++;
          ctr++;
       }
   }
   return (ctr == 0) ? 1 : 0 ;
}
```

TEST 2 - Problem C | **Obtained score:**

```
1   char* checkCreditCards(int card[][16], int numCards){
2      char* ret = (char*)malloc((numCards + 1) * sizeof(char));
3      int x, y, sum;
4      if (ret != NULL){
5         for(x = 0; x < numCards; x++){
6            sum = 0;
7            for(y = 0; y < 16; y++){
8               if(y % 2 == 0){
9                  card[x][y] *= 2;
10                 if(card[x][y] >= 10)
11                    card[x][y] = 1 + card[x][y] % 10;    /* or - 10 */
12              }
13              sum += card[x][y];
14           }
15           ret[x] = (sum % 10 == 0) ? 'V' : 'I';
16        }
17     }
18     ret[x] = -1;
19     return ret;
20  }
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

TEST 2 - Problem D | Obtained score:

```
/*For 10 points*/
void addNewProducts(productList *L, char filename[]){
   FILE* fp;
   fp = fopen(filename,"r");
   if(fp!=NULL){
      while(L->ctr < SIZE && fread(&L->prod[L->ctr], sizeof(products), 1, fp) != 0){
         if(L->prod[L->ctr].exp == NONPERISHABLE){
            L->ctr++;
         }
      }
      fclose(fp);
   }
}

/*For 15 points*/
int removeBrand(productList *L, char brand[]){
   int num = 0, x;
   FILE* fp = fopen("RemovedProducts.dat", "a");
   if (fp != NULL){
      for(x = 0; x < L->ctr; x++){
         if(strcmp(L->prod[x].PName.brand,brand) == 0){
            fwrite(&L->prod[x], sizeof(products), 1, fp);
            num++;
         }
         L->prod[x - num] = L->prod[x];
      }
      fclose(fp);
   }
   L->ctr -= num;
   return num;
}
```