CIS 1201 Mock Final Exam - May 22, 2021

Name:	ANSWER KEY
Group #:	

TEST 1

1.	scanf("%c",*c);
2.	printf("%c",**c);
3.	1
4.	8
5.	20
6-7.	13
8-9.	8
10.	ctype.h
11.	stdio.h
12-14.	if (islower(*str)!=0) *str=toupper(*str);
15-16.	memcpy(Accounts2,Accounts1,10*sizeof(struct user);
17-18.	tempAcc1 = Accounts2[3];
19-20.	strcpy(tempAcc2->username,Accounts2[9].username);
21.	string.h
22.	8084
23.	48
24-25.	25
26-27.	30
28-30.	return ((number & 1 << n - 1) != 0) ? 1 : 0;
31.	2
32.	8
33.	8
34.	12
35-36.	colorlist=(struct color*)malloc(sizeof(struct color));
37-38.	1000
39.	2
40-41.	12
42-43.	0.0, 2.0, 3.0, 6.0, 4.0, 9.0
44-45.	7
46-48.	9,2,9
49-50.	124
51.	15
52.	True
53.	False
54.	-6
55.	2FE030
56.	2FE014
57.	2FE018
58-59.	int(*ptr)[5]=arr;
60-62.	-6
63.	numList*A
64.	numList*B
65.	numList*
66-67.	trav=*B;*trav!=NULL;trav=&(*trav)->nextnum
68.	*trav=*A
69-70.	*A=NULL

```
1    /*For 15 points - use recursion*/
2
3    int computePoly(polynomial poly, int x){
4       if (poly == NULL)
5           return 0;
6       int temp = 1, ctr;
7       for(ctr = poly->expo; ctr > 0; ctr--){
8           temp *= x;
9       }
10      temp *= poly->coe;
11      return temp + computePoly(poly->ptr, x);
12   }
13
14
15   /*For 12 points - without recursion*/
16
17   int computePoly(polynomial poly, int x){
18      polynomial trav;
19      int sum = 0, temp, ctr;
20      for(trav = poly; trav != NULL; trav = trav->ptr){
21          temp = 1;
22          for(ctr = poly->expo; ctr > 0; ctr--){
23              temp *= x;
24          }
25          temp *= poly->coe;
26          sum += temp;
27      }
28      return sum;
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
```

```
1    int checkStrings(twostrings *A){
2        int x, ctr = 0, countArr[26] = {0}; /*can use calloc(), but should not forget to free()*/
3        for(x = 0; A->str1[x] != '\0'; x++){
4            countArr[A->str1[x] - 'A']++;
5            countArr[A->str2[x] - 'A']--;
6        }
7        for(x = 0; x < 26 ; x++){
8            while(countArr[x] < 0){
9                A->extras[ctr] = countArr[x] + 'A';
10               countArr[x]++;
11               ctr++;
12           }
13       }
14       return (ctr == 0) ? 1 : 0 ;
15   }
```

```
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  }
```

```
1   /*For 10 points*/
2   void addNewProducts(productList *L, char filename[]){
3       FILE* fp;
4       fp = fopen(filename,"r");
5       if(fp!=NULL){
6           while(L->ctr < SIZE && fread(&L->prod[L->ctr], sizeof(products), 1, fp) != 0){
7               if(L->prod[L->ctr].exp == NONPERISHABLE){
8                   L->ctr++;
9               }
10          }
11          fclose(fp);
12      }
13  }
14
15  /*For 15 points*/
16  int removeBrand(productList *L, char brand[]){
17      int num = 0, x;
18      FILE* fp = fopen("RemovedProducts.dat", "a");
19      if (fp != NULL){
20          for(x = 0; x < L->ctr; x++){
21              if(strcmp(L->prod[x].PName.brand,brand) == 0){
22                  fwrite(&L->prod[x], sizeof(products), 1, fp);
23                  num++;
24              } else {
25                  L->prod[x - num] = L->prod[x];
26              }
27          }
28          fclose(fp);
29      }
30      L->ctr -= num;
31      return num;
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
```