Problem Solving Final Exam (Model Answer)

Spring (2020/2021)

Question 1:

1. **True (√)**
2. **False (X) → Correction /* */**
3. **False (X)**
4. **False (X) → Correction Sorted Array**
5. **False (X) → i = +1 not valid Correction i += 1**
6. **False (X) → Correction logical expression**
7. **True (√)**
8. **False (X) → Correction infinity times**
9. **True (√)**
10. **False (X) → scanf() will read problem only and neglect anything after space**

Question 2:

1. **c (X <=0 || X >= 100)**
2. **c *Y = 5;**
3. **a 0**
4. **c X[3]+=1**
5. **b void SetVal(int *X);**
6. **a get_max(const int list[ ],int n);**
7. **d (a) , (b) and (c)**
8. **b printf("%-10s", y);**
9. **a mOHAMED aMR**
10. **a strcat**

Question 3:

1. **trace of the program**

program	s	a	i
char s[35];			
char* a;			
int i = 1;			1
strcpy(s, "Introduction to problem solving");	Introduction to problem solving		
printf("%s\n",s);	o/p → Introduction to problem solving then new line		
a = strtok(s, " ");		Introduction	
printf("%s.\n",a);	o/p → Introduction. then new line		

while (i < 3)			1 < 3 true then cond true
a = strtok(s, " ");		Introduction	
printf("%s.\n",a);	o/p → Introduction. then new line for the second time		
i++;			2 Then check 2<3 true then cond true
		Introduction	
	o/p → Introduction. then new line for the third time		
			3 then check 3<3 false So stop the loop

So the final output:

Introduction to problem solving
Introduction.
Introduction.
Introduction.

2. **trace of the program**

Program	a	i	i<5	a[i]	a[i+1]	a[i] %2 == 0	a[i+1] %7==0	i<=5
int a[] = { 10,3,4,7,2,1 };	10,3,4,7,2,1							
int i;								
for (i = 0; i < 5; i = i + 2)		0	true					
a[i] = a[i] + a[i + 1];				10+3=13				
	13,3,4,7,2,1							
a[i + 1] = a[i + 1] * a[i + 1];					3*3=9			
	13,9,4,7,2,1							
if (a[i] % 2 == 0)						13%2 ==0 false		
continue;						no		
if (a[i + 1] % 7 == 0)							9%7 ==0	
break;							no	
	13,9,4,7,2,1							

		2	true					
				4+7=11				
	13,9,11,7,2,1							
					7*7=49			
	13,9,11,49,2,1							
						11%2==0 false		
						no		
							49%7==0 true	
							Yes Break the loop	
for (i = 0; i <= 5; i++)		0						true
printf("%d\n",a[i]);	13,9,11,49,2,1		o/p → print 13 then new line					
		1						true
			o/p → print 9 then new line					
		2						true
			o/p → print 11 then new line					
		3						true
			o/p → print 49 then new line					
		4						true
			o/p → print 2 then new line					
		5						true
			o/p → print 1 then new line					
		6						false stop

So the final output:

13

9

11

49

2

1

3. **trace of the program,**

Program	inp	a	b	c	i	i<=3
FILE* inp;						
int a, b, c, i = 1;					1	
inp = fopen("Data.txt", "r");	Open file Data.txt					
while (i <= 3)						1<=3 true
fscanf(inp, "%d %d %d", &a, &b, &c);		1000	5	1		
printf("%d:%dX%d=%d\n", a, b, c, b * c);	o/p → 1000:5*1=5					
i++;					2	2<=3 true
		1001	4	2		
	o/p → 1001:4*2=8					
					3	3<=3 true
		1003	8	7		
	o/p → 1003:8*7=56					
					4	4<=3 false stop
fclose(inp);	Close the file					

So the final output:

1000:5X1=5

1001:4X2=8

1003:8X7=56

Question4:

1. **the program**

```
#define _CRT_SECURE_NO_WARNINGS

#include<stdio.h>
#include <stdbool.h>

#define size 4

void GetData(int ID[], double GPA[], int* n);

void DataSort(int ID[], double GPA[], int n);
```

```
void DispData(int ID[], double GPA[], int n);

void swapint(int* xp, int* yp);

void swapdouble(double* xp, double* yp);

void main()
{
    int ID[size];
    double GPA[size];
    GetData(ID, GPA, size);
    DataSort(ID, GPA, size);
    DispData(ID, GPA, size);
}

void GetData(int ID[], double GPA[], int* n)
{
    printf("Enter The Data Items of %d Students\n ",n);
    for (int i = 0; i < n; i++)
    {
            printf("\nStudent ID: ");
            scanf("%d",&ID[i]);
            printf("Student GPA: ");
            scanf("%lf", &GPA[i]);
    }
}

void DispData(int ID[], double GPA[], int n)
{
    printf("Student ID \t\t GPA\n");
    for (int i = 0; i < n; i++)
    {
            printf("%d \t\t", ID[i]);
            printf("%lf \n", GPA[i]);
    }
}

void swapdouble(double* xp, double* yp)
{
    double temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void swapint(int* xp, int* yp)
{
```

```
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void DataSort(int ID[], double GPA[], int n)
{
    int i, j;
    bool swapped;
    for (i = 0; i < n - 1; i++)
    {
        swapped = false;
        for (j = 0; j < n - i - 1; j++)
        {
            if (GPA[j] < GPA[j + 1])
            {
                swapdouble(&GPA[j], &GPA[j + 1]);
                swapint(&ID[j], &ID[j + 1]);
                swapped = true;
            }
        }

        // IF no two elements were swapped by inner loop, then break
        if (swapped == false)
            break;
    }
}
```

2. the program

```
#define _CRT_SECURE_NO_WARNINGS

#include<stdio.h>

void main()
{
    int n, sumodd = 0, sumeven = 0, counteven = 0, countodd = 0;
    float avgodd, avgeven;

    for (;;)
    {
        printf("Enter Number: ");
        scanf("%d",&n);
        if (n % 2 == 0)
        {
            sumeven += n;
```

```
                counteven++;
            }
            else
            {
                sumodd += n;
                countodd++;
            }
            if (n == 0)
                break;
    }

    avgeven = sumeven / counteven;
    avgodd = sumodd / countodd;

    printf("\nSum of Even Numbers: %d",sumeven);
    printf("\nAverage of Even Numbers: %f", avgeven);

    printf("\nSum of Odd Numbers: %d", sumodd);
    printf("\nAverage of Odd Numbers: %f", avgodd);
}
```

3. the program

```
#define _CRT_SECURE_NO_WARNINGS

#include<stdio.h>

int decimal_binary_rec(int dec_num)
{
   if (dec_num == 0)
      return 0;
   else
      return (dec_num % 2 + 10 *
         decimal_binary_rec(dec_num / 2));
}

void main()
{
   int num, bin_num;
   printf("Enter Decimal Number to Be Converted: ");
   scanf("%d",&num);
   bin_num = decimal_binary_rec(num);
   printf("\nBinary Number after Conversion: %d \n",bin_num);
}
```