Programming Fundamentals

Fall 2021- Final Exam Solution

General Instructions for marking the paper:

1- **1 point for structure of the program**
2- **1.5 points for declarations of variable and applying loops and decision structures like if, else if and switch.**
3- **2.5 points for taking proper inputs**
4- **2.5 points for logic**
5- **2.5 points for proper output**

Question 1:

MCQs

1.	D
2.	C
3.	D
4.	B
5.	A
6.	D
7.	D
8.	A
9.	A
10.	A

Question 2: Output:

1. ello World World
2. GreenAquaOther
3. mmmmaaaannnn
4. Compilation Error
5. 10 9 10
6. sizeof arri[] = 12 sizeof ptri = 8 sizeof arrc[] = 3 sizeof ptrc = 8
7. 4
8. str1: Hello. How are you?, str2: garbagevalue
9. Runtime Error- Program crashes
10. Error: read-only location ‘*p’

Question 3:

```
#include&lt;stdio.h&gt;
```

DATA INPUT 4 MARKS

```
void get_data(int arr[],int size)
{
int i;
for (i=0; i&lt;size; i++)
{
printf(&quot;Enter numberof card %d :&quot;, i+1);
scanf(&quot;%d&quot;, &amp;arr[i]);
}
}
```

SUM OF DIGITS 4 MARKS

```
int sum_of_digits(int num)
{
if (num == 0)
return 0;
return (num % 10 + sum_of_digits(num / 10));
}
```

DATA SORTING 4 MARKS

```
void sort(int arr[],int size)
{
int i,j;
for (i=0; i&lt;size; i++)
{
for (j=0; j&lt;size-j; j++)
{
if(sum_of_digits(arr[j])&gt;sum_of_digits(arr[j+1]))
{
int temp=arr[j];
arr[j]=arr[j+1];
arr[j+1]=temp;
}
}

}
}
```

DISPLAY 4 MARKS

```
void display(int arr[],int size)
{
int i;
for (i=0; i&lt;size; i++)
{
printf(&quot;\nSorted cards %d : %d&quot;, i+1,arr[i]);
}
}
```

2 MARKS FOR MAIN, DECLARATIOINS, STRUCTURE OR FLOW OF PROGRAM

2 MARKS FOR PROPER OUTPUT

```
int main()
{
int n;
printf(&quot;How many cards you want to enter :&quot;);
scanf(&quot;%d&quot;,&amp;n);
int cards[n];
get_data(cards,n);
sort(cards,n);
display(cards,n);
return 0;
}
```

Question 4: MARKING SCHEME

2 MARKS FOR MAIN, DECLARATIOINS, STRUCTURE OR FLOW OF PROGRAM

2 MARKS FOR PROPER OUTPUT

2 MARKS FOR DO WHILE LOOP

2 MARKS FOR SWITCH CASE

4 MARKS FOR BASIC () FUNCTION IF PROPER IMPLEMENTED ELSE 2 MARKS

4 MARKS FOR DISPLAY () FUNCTION IF PROPER IMPLEMENTED ELSE 2 MARKS

4 MARKS FOR DOT () FUNCTION IF PROPER IMPLEMENTED ELSE 2 MARKS

Question 4:

```
#include<stdio.h>
#define SIZE 5
struct data{
        float op1, op2;
        char opr;};
float basic(float, float, char);
void display(const struct data [], int);
double dot(const double [], const double [], int);
```

```
int main(){
        char choice;
        int i;
        printf("Do you want to solve an equation (press a) or a dot product (press b)? ");
        choice=getche();
        puts("");
        if(choice=='a'){
                struct data d[100];
                int n;
                do{
                i=0;
                printf("\nHow many operands are there in the equation: ");
                scanf("%d",&n);
                printf("Enter operand %d: ",i+1);
                scanf("%f", &d[i].op1);
                        for(i=0;i<n-1;i++){
                                printf("Choose operator(+,-,*,/): ");
                                scanf(" %c",&d[i].opr);
                                printf("Enter operand %d: ",i+2);
                                scanf("%f", &d[i].op2);
                                if(d[i].opr=='/' && d[i].op2==0){
                                        printf("Invalid. Any number divided by zero is
infinity.\n");
                                        break;
                                        }
                                else if(d[i].opr=='+' || d[i].opr=='-' || d[i].opr=='*' || d[i].opr=='/')
                                        d[i+1].op1=basic(d[i].op1, d[i].op2, d[i].opr);
                                else{
                                        printf("Invalid. please choose an operator from the
given list.\n");
                                        break;}
                                }
                                if(i==n-1)
                                        display(d,n);
                                printf("Do you want to solve another equation? Press y for yes!");
                        }while(getche()=='y');
        }
        else if(choice=='b'){
                double A[SIZE], B[SIZE];
                do{
                        puts(" ");
                        for(i=0;i<SIZE;i++){
                                printf("Enter the value for A[%d] and B[%d]: ", i, i);
                                scanf("%lf %lf", &A[i], &B[i]);
                        }
                        printf("A . B = %lf\n",dot(A,B, SIZE));
                        printf("Do you want to perform another dot product? Press y for
yes!");
                }while(getche()=='y');
        }
        else
                printf("You did not choose one of the given options.");


        return 0;
}


float basic(float n1, float n2, char op){
        switch(op){
                case '+':
                        return n1+n2;
                case '-':
                        return n1-n2;
                case '*':
                        return n1*n2;
                case '/':
                        return n1/n2;
        }
}
void display(const struct data m[], int n){
        int i=0;
        printf("Call to display function: You have solved the following equation.\n");
        printf("%0.2f",m[i].op1);
        for(i=0;i<n-1;i++){
                printf(" %c %0.2f",m[i].opr,m[i].op2);
        }
        printf(" = %0.2f\n",m[i].op1);
}
double dot(const double M[], const double N[], int s){
        int i;
        double res=0;
        for(i=0;i<s;i++)
                res+=M[i]*N[i];
        return res;
}
```

Question 5. Part a.

1 MARK TO ATTEMPT THE QUESTION

1 MARKS FOR FUNCTION DECLARATION

1 MARK FOR RECURSIVE CALL

2 MARKS FOR CORRECT LOGIC

```
void rev(char [],int);
int main(){
          int i=15;
          char s[15];
          strcpy(s,"this is morning");
           rev(s,0);


}
void rev(char s[15],int i){


    if (s[i] != '\0') {
        rev(s,++i);
        printf("%c",s[i]);
    }
}
```

Question 5. Part b.

1 MARK TO ATTEMPT THE QUESTION

1 MARKS FOR PROPER FUNCTION DECLARATION

2 MARKS FOR CORRECT LOGIC

1 MARK FOR CORRECT OUTPUT

merge (char s1 [ ], int sp, char s2[“text to be insert”])

```
1-  Set i=0,j=0
2-  length = s1.length + s2.length
3-  declare new array[length]
4-  for(i=0; i<sp; ++i){
            s3[j]=s1[i];
            j++;
    }
5-  for(i=0; i<s2.length; ++i){
            s3[j]=s2[i];
            j++;
    }
6-  for(i=sp-1; i<s1.length; ++i){
            s3[j]=s1[i];
            j++;
    }
7-  print s3.
```

NOTE:

It is not necessary that the student uses pointers to accept and return the string, But if he uses pointers no extra marks will be given. The student may use a very simple logic to perform the task.

Question 5. Part c.

1 MARK TO ATTEMPT THE QUESTION

1 MARKS FOR PROPER FUNCTION DECLARATION

2 MARKS FOR CORRECT LOGIC / OR USING RECURSIVE CALL

1 MARK FOR CORRECT OUTPUT

Question 5. Part c.

```
bool compare ( char p[], char s[],int i, int j)
{       if (p[ i ] == ‘\0’ && s[ j ] == ‘\0’)
                return (true);
```

```
        if (p[ i ] == ‘?’ || p[ i ]==s[ j ])
                return compare (p,s,++i, ++j);
        if(p[ i ] == ‘*’)
                return compare (p,s,++i,j) || compare (p,s,i,++j);
        return (false);
}
```

NOTE:

The student can use pointers as well.

Question 5. Part d.

1 MARK TO ATTEMPT THE QUESTION

1 MARKS FOR PROPER FUNCTION DECLARATION

2 MARKS FOR CORRECT LOGIC

1 MARK FOR CORRECT OUTPUT

Question 5. Part d.

```
#include <stdio.h>

void recurse();
void main()
{
char arr[]={"this is the test string"};
recurse(arr,0);
}
void recurse(char a[], int n)
{
int st=n;
int i;
int en=st;
while(1)
{
 if(a[en]=='\0')
  break;
 else if(a[en] == ' ')
 {
  recurse(a,en+1);
  break;
 }
 else
  en++;
}
for(i=st;i<=en;i++)
{printf("%c",a[i]);
if(a[i]=='\0')
printf(" ");

}
return;
```

Question 6:

Attempt (declarations, flow, and structure of program): 2.5

Matrix Initialization: 2.5

Second Matrix Jammer initialization: 2.5

Calculating Transpose: 5

Printing the transpose: 2.5

Adding two matrices a and b: 2.5

Result storing in matrix result and printing the result: 2.5

MODEL SOLUTION

~~~~~~

```
#include <stdio.h>

#define MAXROW 3

#define MAXCOL 4

/*User Define Function to Read Matrix*/

void readMatrix(int m[][MAXCOL],int row,int col)

{

int i,j;
```
~~~~~~

```
for(i=0;i< row;i++)
{
for(j=0;j< col;j++)
{
printf("Enter element [%d,%d] : ",i+1,j+1);
scanf("%d",&m[i][j]);
}
}
}
/*User Define Function to Read Matrix*/
void printMatrix(int m[][MAXCOL],int row,int col)
{
int i,j;
for(i=0;i< row;i++)
{
for(j=0;j< col;j++)
{
printf("%d\t",m[i][j]);
}
printf("\n");
}
}
int main()
{
int a[MAXROW][MAXCOL],b[MAXROW][MAXCOL],result[MAXROW][MAXCOL];
int i,j,r1,c1,r2,c2,sum,avg,transpose[10][10];

printf("Enter number of Rows of matrix a: ");
```

```
scanf("%d",&r1);
printf("Enter number of Cols of matrix a: ");
scanf("%d",&c1);
printf("\nEnter elements of matrix a: \n");
readMatrix(a,r1,c1);
// SECOND MATRIX
printf("Enter number of Rows of matrix b: ");
scanf("%d",&r2);
printf("Enter number of Cols of matrix b: ");
scanf("%d",&c2);
printf("\nEnter elements of matrix b: \n");
readMatrix(b,r2,c2);
// TRANSPOSE OF A MATRIX
// computing the transpose
for (i = 0; i < r2; ++i)
for (j = 0; j < c2; ++j) {
transpose[j][i] = b[i][j];
}
// printing the transpose
printf("\nTranspose of the matrix:\n");
for (i = 0; i < c2; ++i)
for (j= 0; j < r2; ++j) {
printf("%d ", transpose[i][j]);
if (j == r2 - 1)
printf("\n");
}
/*sum and sub of Matrices*/
if(r1==r2 && c1==c2)
```

```
{
/*Adding two matrices a and b, and result storing in matrix result*/
for(i=0;i< r1;i++)
{
for(j=0;j<c1;j++)
{
result[i][j]=a[i][j]+b[i][j];
}
}
/*print matrix*/
printf("\nMatrix after adding (result matrix):\n");
printMatrix(result,r1,c1);
// ADDING ALL ELMENETS
int sum;
sum=0;
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
{
sum += result[i][j];
}
}
printf("\nSUM of all elements : %d \n",sum);
avg = sum/12;
printf("\nAverage of all elements : %d \n",avg);
if (avg>=15)
printf("Abnormal Situation");
else
printf("Normal Situation");
}
else
{
printf("\nMatrix can’t be added, Number of Rows & Cols are Different");
}
return 0;
```