National University of Computer & Emerging Sciences, Karachi
Fall-2022 School of Computing
Final Examination Solution
19th Dec 2022, 01:00 pm – 04:00pm

Course Code:CS1002	**Course Name:** Programming Fundamentals
Instructor Name: Mr. M. Shahzad, Dr. Farooque, Dr. M. Farrukh Shahid, Dr. M. Usama, Mr. Shahbaz, Mr. Musawar, Ms. Atiya, Ms. Aqsa, Ms. Sumaiyah	
Student Roll No:	**Section:**

Instructions:

- Return the question paper and make sure to keep it inside your answer sheet.
- Read each question completely before answering it. There are total **four questions on four printed sides of two pages**.
- In case of any ambiguity, you may make assumptions. However, your assumption should not contradict any statement in the question paper.
- Do not write anything on the question paper (except your ID and section). You will be graded ONLY on answer sheet.

Total Time: 3 Hour **Max Points**: **100**

Question # 1 **[40 points (4 each), 70 mins] CLO1**

A. Considering the following programs and illustrate the required process in graphical form. Assume all necessary header files are included and all programs are syntactically correct.

a.

By using malloc

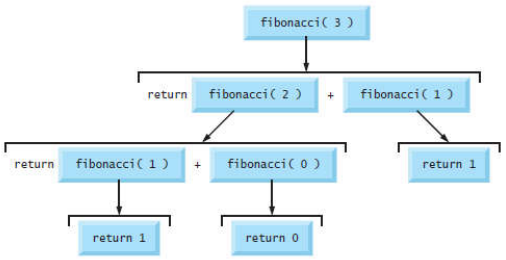
Garbage Value	Garbage Value	Garbage Value	Garbage Value	Garbage Value
64230	64238	64246	64254	64262

By using Calloc

0	0	0	0	0
64230	64238	64246	64254	64262

b. Draw the recursive stack of the following function, if we call sum(3) with n = 3. [SIMILAR TO BELOW]

c. Draw



d. Draw

Address	64230	64234	64238	64242	64252	64256	64260
Variable	id1	id2	birthday.date	birthday.month	birthday.year	a	percentage

B. Considering the output, write down the missing part of the program. You must write only the missing part on the answer sheet with the most appropriate code. **CLO2**

a.
```
#include <stdio.h>
typedef struct{
    int id; float price; char name[20];
}userTyped;
void main() {
    userTyped inst1[]={{20, 5000.05, "Samsung"},
                       {30, 3300.25, "Apple"},
                       {40, 6020.05, "Acer"}};
    userTyped *ptr = inst1;
    //Using variable
    for(int i = 2; i >= 0; i--)
        printf("%d, %f, %s\n", inst1[i].id, inst1[i].price,
inst1[i].name);
    printf("------------\n");
    //Using pointer
    for(int i = 0; i < 3; i++)
        {
            printf("%d, %f, %s\n", ptr->id, ptr->price, ptr->name);
            ptr++;

        }
}
```

Output:
40, 6020.0, Acer
30, 3300.2, Apple
20, 5000.0, Samsung

20, 5000.0, Samsung
30, 3300.2, Apple
40, 6020.0, Acer

b.
```
void main(){
char country[] = "Pakistan";
void *ptr;
ptr = country;
while( *((char*)(ptr)) != '\0')
  {
  printf("%c", *((char*)(ptr)));
  ptr++;
   }}
```

Output:
Pakistan

c.
```
void main(){
 char ch, *str;
 int cnt=0;
 puts("enter any string: ");
 while((ch=getche()) != 13){
   if(cnt==0){
 str = (char *) malloc (sizeof(char));
 str[cnt]=ch;}
   else{
       str = (char*) realloc(str, (cnt+2)*sizeof(char));
       str[cnt] = ch;

       }
       cnt++;
       }
 str[cnt]='\0';
 printf("\n%s",str);
}//Hint: You need to extend the dynamic array in this problem
```

Output:
It will produce **"Pakistan Zindabad"** if input is **"Pakistan Zindabad"**

d. Initialize and display the record structure:
```
struct employee{
    int eid;  char ename[20];
};
struct date{
   int joiningYear;};
struct record{
  struct employee emp;     struct date dt;
};
void main(){
       struct record rcd[2]={
                {{101,"Asad"}, 2010},
                 {{102,"Bilal"}, 2014}};
    int i;
    for(i  = 0; i <2; i++){
       printf("Employee ID: %d \nName: %s \nJoining Year: %d\n\n",
rcd[i].emp.eid, rcd[i].emp.ename, rcd[i].dt.joiningYear);
       }
}
```

Output:
Employee ID:101
Name: Asad
Joining Year: 2010

Employee ID: 102
Name: Bilal
Joining Year: 2014

Output:

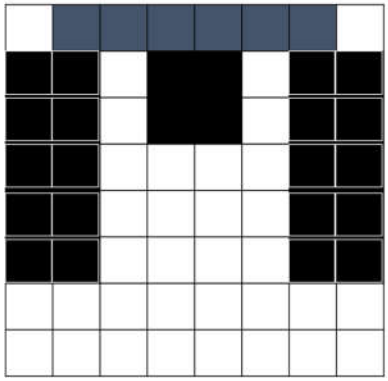
`#include <stdio.h>`
`void main() {`
`  int arrAll[] = {80, 82, 79, 71, 82, 65, 77};`
`  for(int j = 0; j < 7; j++)`
`  {`
`  for(int i = 0; i <= j; i++)`
`  printf("%c", arrAll[i]);`
`  printf("\n");`
`  }`
`}`
	P
	PR
	PRO
	PROG
	PROGR
	PROGRA
	PROGRAM

e.
`void main(void){`
`  char *p[3] = {"Rashid", "Sajid", "Ali",};`
`  char * tmp;  int i, j;`
`for(j=0; j<3-1-i; j++){`
`  if(strcmp(p[j], p[j+1]) > 0){`
`  //swap array[j] and array[j+1]`
`  strcpy(tmp, p[j]);`
`  strcpy(p[j], p[j+1]);`
`  strcpy(p[j+1], tmp);`
`  }`
`  }`
`  }`
`  for( i = 0; i<3; i++)`
`  puts(p[i]);`
`  }`
	Output:
	Ali
	Rashid
	Sajid

Question#2: **[12 points (4 each), 20 mins] CLO2**

A 2D picture array contains data representing a bitmap image. Each element of the array represents a pixel of the image. The image is grayscale encoded where the values of each pixel range from 0 (representing black) to 255 (representing white), with intermediate values representing different levels of gray. The following is an **example** of an image and the corresponding data values for the picture array.

Bitmap Image



Values

240	10	10	10	10	10	10	240
80	80	240	80	80	240	80	80
10	10	240	10	10	240	10	10
10	10	240	240	240	240	10	10
10	10	240	240	240	240	10	10
10	10	240	240	240	241	10	10
150	240	150	240	150	240	150	240
150	240	150	240	150	240	150	240

A method, Lighten(), is required to lighten the image. Lightening an image may cause it to "burnout". An image is said to be "burnt out if any pixel is set to the maximum value 255.
The function Lighten() will:

- Increase the value of each pixel by 10%.
- Return 1 if the resultant image is 'burn out', else 0.

a) Implement the Lighten() function.
b) Implement a function display(), which displays the values of the matrix after implementing the Lighten() method.
c) Your program should take initial inputs for all pixels in M x N matrix while handling odd inputs / exceptions. Exception is a case where the entered pixel value is less than 0 and greater than 255.

SOLUTION:

Code:

```
#include<stdio.h>
int lighten(int image[3][3], int row, int col){
	int rowCtr, colCtr;
	for (rowCtr = 0; rowCtr < row; rowCtr++){
		for(colCtr = 0; colCtr < col; colCtr++){
				image[rowCtr][colCtr] *= 1.10;
				if(!(image[rowCtr][colCtr] >= 0 &&
image[rowCtr][colCtr] <= 255))
					return 1;
		}
	}
	return 0;
}
void display(int image[3][3], int row, int col){
	puts("\nDisplaying the matrix after lightening");
	int rowCtr, colCtr;
	for (rowCtr = 0; rowCtr < row; rowCtr++){
		for(colCtr = 0; colCtr < col; colCtr++){
			printf("%d ", image[rowCtr][colCtr]);
		}
		puts("");
	}

}

int main(){
	int row, col, rowCtr, colCtr;
```

```
        puts("Enter the number of rows and cols");
        scanf("%d %d", &row, &col);
        int image[row][col];
        for (rowCtr = 0; rowCtr < row; rowCtr++){
                for(colCtr = 0; colCtr < col; colCtr++){
                        do{
                                printf("Enter the row %d and col %d : \n",
rowCtr, colCtr);
                                scanf("%d", &image[rowCtr][colCtr]);
                        }
                        while(!(image[rowCtr][colCtr] >= 0 &&
image[rowCtr][colCtr] <= 255));
                }
        }
        if(lighten(image, row, col))
                puts("Image is burnt out");
        else
                display(image, row, col);
}
```

Question # 3 **[24 points (6 each), 50 mins]**
CLO3

Suppose that you are required to develop Account Management System for a Car's Show Room to calculate overall tax, retail price (Selling price to customer that include GST) and sum of profit from the sales. All cars have 15% import duty tax from ***Capital Cost*** (Cost that seller buy from manufacturer) that needs to be paid to Pakistan Custom. A luxury car has 10% sales tax, and a non-luxury car has 6.5 % sales tax from the capital cost that need to pay. The seller needs 75% of retail profit from all total cost (include the cost of import and sales tax) per car either luxury or non-luxury car. Lastly, 6% Good and Services Tax (GST) is added to the cost price that will become the retail price for a car. For all cars, customers need to register name, address.

Hint: First formulate how to calculate car import duty tax, luxury car sales tax, total profit from capital cost and tax, total price for luxury car, total services tax for any good as per Pakistan customs, and retail sales price of luxury and non-luxury car (price include GST).

a. Write a program based on the following specifications:
    I. Develop a structure **CustomeInfo** to the following specification: The structure has two instance members **Name, Address.**
    II. Develop a structure **Car** to the following specification: The structure has five data members as, **price, Model, Brand, ManufacturingDate, CountryOfOrigin and CustomerInfo.**
b. ***SaveBillinfo*** function **gets** input from user for customer and car, **stores** in structures, and **saves** customer bill (customer and car info) to ***bill.dat*** file for a specific customer importing specific car.
c. ***GetBillinfo*** function **opens** text file in binary format, **reads** data from File, and **prints** on screen.
d. ***PrintAllwithTaxDetails*** function **displays** customer information, billing Information along with the tax, and net profit details. It must call following functions to print all details of particular car. You must implement these functions as well.
    - ***ServicesTax*** function calculates the service tax and returns ServicesTax
    - ***RetailProfit*** function calculates the Retail Price and returns RetailProfit
    - ***ImportDutyTax*** function calculate the importDutyTax and returns importDutyTax
    - ***SalesTax*** function calculates the sales tax and returns SalesTax
    - ***CalulatePrice*** function calculates the price after sales tax, import tax, GST and net profit and returns Calculated Price.

SOLUTION:

Code:

```
#include <stdio.h>
#include <String.h>

        FILE *outFile;

        struct CustomerInfo{
                char CustomerName[50];
                char AddressName[50];
        };

        struct Car {
                int Price;
                int Model;
                char Brand[50];
                char ManufacturingDate[50];
                char CountryOfOrigin[50];
                struct CustomerInfo CI;
        };
```

```
	void printline() {
		printf("\t ---------------------------------------- \n");
	}

	long ServicesTax(int Price) {
		return (Price * 75) / 100;
	}

	long RetailProfit(int Price) {
		return (Price * 75) / 100;
	}

	long importDutyTax(int Price) {
		return (Price * 15) / 100;
	}

	long SalesTax(int Price) {
		return 10 * Price/ 100;
	}

	long CalulatePrice(int Price) {

		long temp = SalesTax(Price) + ServicesTax(Price) + RetailProfit(Price) +
importDutyTax(Price);
		temp += Price;
		return temp;
	}

	void PrintAllDetails(struct Car c) {
		printf("\tCustomer Name:\t\t %s \n",c.CI.CustomerName);
		printf("\tCustomer Address:\t %s \n",c.CI.AddressName);
		printline();
		printf("\t\tCar INFORMATION \n");
		printline();
		printf("\tManufacturing Year:\t %s \n",c.ManufacturingDate);
		printf("\tCar Price:\t\t %d \n",c.Price);
		printf("\tCar Brand:\t\t %s \n",c.Brand);
		printf("\tCar Model:\t\t %d \n",c.Model);
		printf("\tCar Country of Origin:\t %s \n",c.CountryOfOrigin);
		printline();
		printf("\t\tBILLING DETAILS \n");
		printline();
		printf("\tImport Duty Cost: \tRs %ld \n",importDutyTax(c.Price));
		printf("\tSales Tax Cost: \tRs %ld \n",SalesTax(c.Price));
		printf("\tRetail Price: \t\tRs %ld \n",RetailProfit(c.Price));
		printline();
		printf("\tFinal Price: \t\tRs %ld \n",CalulatePrice(c.Price));
		printf("\t********THANKYOU FOR SHOPPING. **********\n\n\n");
	}

	void printBill(struct Car c){
		printf("\t\tEnter CUSTOMER INFORMATION \n");
```

```
printline();

printf("\tCustomer Name:\t\t");
scanf("%s",&c.CI.CustomerName);
getchar();

printf("\tCustomer Address:\t");
scanf("%s",&c.CI.AddressName);

printf("\tManufacturing Year:\t");
scanf("%s",&c.ManufacturingDate);

printf("\tCar Price:\t\t");
scanf("%d",&c.Price);

printf("\tCar Model:\t\t");
scanf("%d",&c.Model);
```

```
		printf("\tCar Brand:\t\t");
		scanf("%s",&c.Brand);

		printf("\tCar Country of Origin:\t");
		scanf("%s",&c.CountryOfOrigin);

		PrintAllDetails(c);

		fwrite(&c, sizeof(struct Car),1,outFile);

	}

	void ReadData(){
		struct Car obj;
		printf("\n\n\t\tCUSTOMER INFORMATION");
		printline();
		rewind(outFile);
		while(fread(&obj, sizeof(struct Car),1,outFile)){
			PrintAllDetails(obj);
		}
	}

	void NetProfit(){
		FILE *inFile;
		fopen("Car.dat", "w+");
		struct Car obj;

		long netprofit=0;
		while(fread(&obj, sizeof(struct Car),1,inFile)){
			netprofit =netprofit + CalulatePrice(obj.Price);
		}
		fclose(inFile);
		printf("\n\n\t\tCUSTOMER INFORMATION \n");
		printline();
		printf("%ld",netprofit);
	}

	int main(){
		int choice =0;
		outFile = fopen("Car.dat", "w+");
		do{
			printf("\n\n\tENTER CHOICE\n \t1.PERFORM A
TRANSECTION\n\t2.DISPLAY ALL THE TRANSECTIONS\n\t3.DISPLAY NET Profit\n");
			printf("\tMake a choice: ");
			fflush(stdin);
			scanf("%d",&choice);
```

```
		//printf("\n\n\n Checking %d \n\n\n",choice);
		system("CLS");

		if(choice==1){
			//system("CLS");
			struct CustomerInfo CI = {"M Ali", "Block C , Karachi"};
			struct Car test = {445,2016, "Honda","2016","Japan",CI};
			printBill(test);
		} else if(choice==2){
			ReadData();
		} else if (choice==3)
			NetProfit();
	}while(choice!=0);
	fclose(outFile);
	return 0;
}
```

Question # 4 **[24 points (6 each), 50 mins]**
CLO4

You are required to develop a system in C language to keep track of all participating groups in the "Coder's Cup" competition. Each group is assigned a set number of tasks. The program intends to include the following modules. The solution must be provided using only the mentioned functions. Global variables are not allowed. You must use appropriate data types, return types and function arguments.

a) Module Name: Input().The working of this module goes as follows:
   i- The ***Input()*** function is called whenever user wishes to add new participating group information (GroupID, GroupName, 5 tasks results).
   ii- Each task's value must be entered in the form of 0's or 1's. If any other value is entered, the program must prompt to re-enter.
      a. 0 means that the group become unsuccessful at solving a particular task
      b. 1 means that the group become successful at solving a particular task
   iii- Append the newly taken data into the file named as **CompRecord.txt**.
   iv- The **CompRecord.txt** file will have set of records where each record contains participating group's complete information.

b) Module Name: **DisplayWinner()**. This module finds and prints all winning groups information. Any group is considered to be a winner who has majority of successful attempted all tasks. There can be multiple winners.

c) Module Name: **Search()**.This module displays the status/ details of any given group. The user must be allowed to search until he/she enters **0**. For example, if user enters **3**, the data against GroupID = 3 must be displayed that shows GroupName, and its successful and failed tasks.

HINT:

- You are allowed to define parameters and return types of these functions as you find appropriate.
- All above modules are dealing with the data stored in the file.

SOLUTION:

Code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void input(void);

int DisplayWinner(int,int,int*);
int DisplayLoser(int,int,int *);
void Search(int,int,int*);

void main(void)

{

        input();

}

void input(void)

{

        int i,chal,choice,group,j;

        printf("Enter Total Challenges : ");
        scanf("%d",&chal);
```

```
printf("Enter Total Groups : ");
scanf("%d",&group);

int*resultGrid;

	resultGrid=(int*)malloc(chal*group*sizeof(int));
int k=-1;

printf("\n");
for(i=0;i<group;i++)

{

	for(j=0;j<chal;j++)

	{

		k++;

		printf("Enter Group %d\'s Task %d Score(1 or 0) : ",i+1,j+1);

		scanf("%d",&*(resultGrid+k));

		if((*(resultGrid+k))==0||(*(resultGrid+k))==1)

		{


		}

		else

		{	j--;


		}

	}


	printf("\n");

}

printf("Enter 1 to display winner.\n");
printf("Enter 2 to display loser.\n");

printf("Enter 3 to search.\n");
printf("Choice : ");
scanf("%d",&choice);

if(choice==1)

{

	printf("Winner is Group : %d",DisplayWinner(chal,group,resultGrid));

}else if(choice==2)

{
```

```
                printf("Loser is Group : %d",DisplayLoser(chal,group,resultGrid));
        }
        else if(choice==3)
        {
                Search(chal,group,resultGrid);
        }


}
int DisplayWinner(int chal,int group,int *resultGrid)
{
        int i,j,k=-1,total[group];
        for(i=0;i<group;i++)
        {
                total[i]=0;
                for(j=0;j<chal;j++)
                {
                        k++;
                        if((*(resultGrid+k))==1)
                        {
                                total[i]=total[i]+1;
                        }
                }
        }
        int max=0,win;
for(i=0;i<group;i++)
```

```
{
        if(total[i]>max)
        {
                max=total[i];
                win=i;
   }
}
        return win+1;
}
int DisplayLoser(int chal,int group,int *resultGrid)
{
        int i,j,k=-1,total[group];
        for(i=0;i<group;i++)
        {
                total[i]=0;
                for(j=0;j<chal;j++)
                {
                        k++;
                        if((*(resultGrid+k))==1)
                        {
                                total[i]=total[i]+1;
                        }
                }
        }
        int least=99999,lose;
for(i=0;i<group;i++)
{
        if(total[i]<least)
        {
                least=total[i];
                lose=i;
   }
}
```

```
        return lose+1;

}

void Search(int chal,int group,int *resultGrid)

{

        int i,n,count=0;
        while(1)

        {

        printf("Enter Group Number : ");
        scanf("%d",&n);

        if(n==0)
        exit(0);
        else

        {

        int k=(chal*(n-1));
        for(i=k;i<k+chal;i++)


        {

                count++;

                printf("Group %d Task %d = %d\n",n,count,*(resultGrid+i));

        }

        Search(chal,group,resultGrid);

   }

}}
```

***** Best of Luck *****

"Programming is a skill best acquired by practice." Alan Turing