2011358042

CSE 115, Programming Language I
Final Exam, Summer 2020
Total 40

1. (5 Points) Write a recursive function to calculate the summation 1+2+3+...+n

```
#include<stdio.h>
int nsum(int n)
{
    if (n==0)
    return 0;
    else
    return (n+nsum(n-1));
}
int main ()
{
    int n, sum;
    printf("Finding out the summation of 1+2+3+...+n \n");
    printf("Enter n: ");
    scanf("%d",&n);
    sum = nsum(n);
    printf("Sum=%d",sum);
    return 0;
}
```

2. (7 Points) Write C program to insert 35 at index 3 of the array {23, 12, 19, 12, -13}. The resultant array will be {23, 12, 19, 35, 12, -13}.

```
#include<stdio.h>
int main()
{
    int arr[10]={23,12,19,12,-13};
    int i,n;
    printf("The array is: { ");
    for(i=0; i<5; i++)
    {
        printf("%d", arr[i]);
    }
    printf("}");
    for(i=5; i>=4; i--)
    {
        arr[i]=arr[i-1];
    }
    printf("\n\nInsert 35 at index 3 of the array\n\n");
    scanf("%d", &n);
    printf("\n\n");
    arr[3]=n;
    printf("The resultant array is: { ");
    for(i=0; i<=5; i++)
    {
        printf("%d", arr[i]);
    }
    printf("} \n\n");
    return 0;
}
```

2011358042

3. (7 points) Write a C code segment to replace all the negative elements in 3X3 2D array with 0. An example

$$\begin{bmatrix} 12 & -3 & 23 \\ -5 & 4 & 2 \\ 11 & -7 & 5 \end{bmatrix} \text{ will be } \begin{bmatrix} 12 & 0 & 23 \\ 0 & 4 & 2 \\ 11 & 0 & 5 \end{bmatrix}$$

```
#include<stdio.h>
int main()
{
    int arr[3][3],i,j;
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            printf("Mat%d%d:", i+1,j+1);
            scanf("%d", &arr[i][j]);
        }
    }
    printf("\n\nThe Matrix is:\n\n");
    for(i=0; i<3 ; i++)
    {
        for(j=0 ; j<3 ; j++)
        {
            printf("%d \t", arr[i][j]);
        }
        printf("\n\n\n");
    }
    printf("\n\n Result:\n\n");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            if(arr[i][j]<0)
                arr[i][j]=0;

            printf("%d\t",arr[i][j]);
        }
        printf("\n\n\n");
    }
    return 0;
}
```

4. (7 points) Write a function void toLowercase(char *s) that converts all the uppercase letters of the string s into lowercase letters.

```
#include<stdio.h>
#include<string.h>
void toLowercase(char *s)
{
    int i;
    for(i=0 ; i<=strlen(s); i++)
    {
        if((s[i] >=65)&&(s[i]<=92))
        {
            s[i] = s[i] +32 ;
        }
    }
    puts(s);
}
int main(void)
{
    char string[100];
    printf("Enter a string: ");
    gets(string);
    printf("\n Lowercase letters:");
    toLowercase(string);
    return 0;
}
```

2011358042

5. (7 points) Define a structure that contains the attributes of a vehicle (for example: color, mode_name, mode_year, etc.). Write codes to take input from user for all the fields in the structure and display the fields.

```
#include<stdio.h>
struct vehicle
{
    char company[50], country[50];
    char color[50], model_name[50];
    int model_year, mileage, price;
};
int main()
{
    struct vehicle vehicle;
    printf("Enter COMPANY of the vehicle: ");
    gets(vehicle.company);
    printf("\n Enter the MODEL NAME of the vehicle: ");
    gets(vehicle.model_name);
    printf("\n Enter COUNTRY of ORIGIN: ");
    gets(vehicle.country);
    printf("\n Enter the COLOR: ");
    gets(vehicle.color);
    printf("\n Enter MODEL YEAR: ");
    scanf("%d", &vehicle.model_year);
    printf("\n Enter the MILEAGE: ");
    scanf("%d", &vehicle.mileage);
    printf("\n Enter the PRICE: ");
    scanf("%d", &vehicle.price);

    printf("\n\n\n Company of the vehicle is: %s\n\n",
                                        vehicle.company);
    printf("Model Name of the vehicle is: %s\n\n", vehicle.model_name);
    printf("Country of origin: %s\n\n", vehicle.country);
    printf("Color: %s\n\n", vehicle.color);
    printf("Model year: %d\n\n", vehicle.model_year);
    printf("Mileage: %d kM\n\n", vehicle.mileage);
    printf("Price: %d BDT\n\n", vehicle.price);

    return 0;
}
```

20113580 42

6. (7 points) Write a program that opens a file called "input.txt", reads one line of string from it, and then adds the string to the content of another file called "output.txt" at the end.

```
#include<stdio.h>
int main()
{
    FILE *fileA, *fileB;
    char str[400];

    fileA=fopen("input.txt", "r");
    fileB=fopen("output.txt", "a");

    if((fileA!=NULL)&&(fileB!=NULL))
    {
        printf("Files found. Press Enter to add the line in
                                        'output.txt' file\n\n\n");
        fgets(str, 400, fileA);
        fprintf(fileB, "%s", str);
    }
    else
    {
        printf("File not found");
    }

    fclose(fileA);
    fclose(fileB);
    return 0;

}
```