Advance ‘C’ programming

Assignment 1:Simple pointers in c.

Set A:

1. Write a program to read two integers using pointers and perform all arithmetic operations on them.

```
#include <stdio.h>
void performArithmetic(int num1, int num2) {
   printf("Sum: %d\n", num1 + num2);
   printf("Difference: %d\n", num1 - num2);
   printf("Product: %d\n", num1 * num2);
   if (num2 != 0)
      printf("Quotient: %d\n", num1 / num2);
   else
      printf("Cannot divide by zero\n");
}

int main() {
   int num1, num2;
   int *ptr1 = &num1;
   int *ptr2 = &num2;

   printf("Enter first integer: ");
   scanf("%d", ptr1);

   printf("Enter second integer: ");
   scanf("%d", ptr2);

   printf("Performing arithmetic operations:\n");
   performArithmetic(num1, num2);
```

```
    return 0;
}
```

```
Enter first integer: 2
Enter second integer: 4
Performing arithmetic operations:
Sum: 6
Difference: -2
Product: 8
Quotient: 0
```

2. Write a program to accept an integer using pointer and check whether it is even or odd

```
#include <stdio.h>
void checkEvenOdd(int *num) {
    if (*num % 2 == 0)
        printf("%d is even.\n", *num);
    else
        printf("%d is odd.\n", *num);
}

int main() {
    int num;
    int *ptr = &num;

    printf("Enter an integer: ");
    scanf("%d", ptr);

    checkEvenOdd(ptr);

    return 0;
}
```

```
Enter an integer: 6
6 is even.
```

```
Enter an integer: 3
3 is odd.
```

3. Write a program to find maximum from two integers using pointers.

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

```
int findMax(int *num1, int *num2) {
   return (*num1 > *num2) ? *num1 : *num2;
}

int main() {
   int num1, num2;
   int *ptr1 = &num1;
   int *ptr2 = &num2;

   printf("Enter the first integer: ");
   scanf("%d", ptr1);

   printf("Enter the second integer: ");
   scanf("%d", ptr2);

   int max = findMax(ptr1, ptr2);
   printf("Maximum value: %d\n", max);

   return 0;
}
```

```
Enter the first integer: 10
Enter the second integer: 6
Maximum value: 10
```

4. Write a program to display the elements of an array containing n integers in the reverse order using a pointer to the array.

```
#include <stdio.h>

void displayReverse(int *arr, int n) {
   printf("Elements of the array in reverse order:\n");
   for (int i = n - 1; i >= 0; i--) {
      printf("%d ", *(arr + i));
   }
   printf("\n");
}

int main() {
```

```
    int n;
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d integers:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    displayReverse(arr, n);

    return 0;
}
```

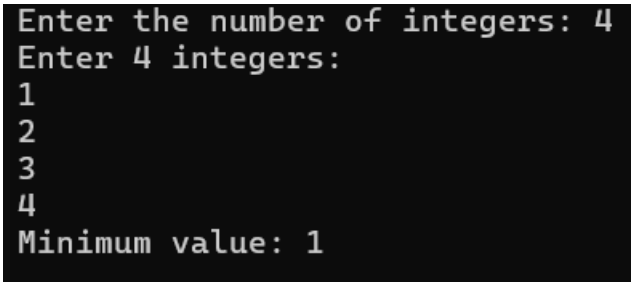
```
Enter the number of elements in the array: 6
Enter 6 integers:
5
4
2
1
3
6
Elements of the array in reverse order:
6 3 1 2 4 5
```

Set B:

1. Write a program to read N integers in an array using pointers and display minimum from them.

```
#include <stdio.h>

void displayMinimum(int *arr, int n) {
    int min = *arr;


    for (int i = 1; i < n; i++) {
        if (*(arr + i) < min) {
            min = *(arr + i);
```

```
        }
    }

    printf("Minimum value: %d\n", min);
}

int main() {
    int n;
    printf("Enter the number of integers: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d integers:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    displayMinimum(arr, n);

    return 0;
}
```



2. Write a function that takes radius of as parameter and two variables. Set first variable to area of circle and second to perimeter of circle. Accept radius in main and also display area and perimeter in main using the above function.

```
#include <stdio.h>

void calculateAreaAndPerimeter(float radius, float *area, float
*perimeter) {
    *area = 3.14159 * radius * radius; // Area of circle formula: p * r^2
```

```
    *perimeter = 2 * 3.14159 * radius; // Perimeter of circle formula: 2
* p * r
}

int main() {
    float radius, area, perimeter;

    // Accept radius from the user
    printf("Enter the radius of the circle: ");
    scanf("%f", &radius);

    // Calculate area and perimeter using the function
    calculateAreaAndPerimeter(radius, &area, &perimeter);

    // Display area and perimeter
    printf("Area of the circle: %.2f\n", area);
    printf("Perimeter of the circle: %.2f\n", perimeter);

    return 0;
}
```

```
Enter the radius of the circle: 6
Area of the circle: 113.10
Perimeter of the circle: 37.70
```

3. Write a function which takes distance in kilometer, centimeter and millimeter as parameters as an integer d and increments the distance by d millimeters. Accept distance and d in main and Display the new distance in main using the above function.

```
#include <stdio.h>

void incrementDistance(int *kilometers, int *centimeters, int
*millimeters, int d) {
    // Increment distance by d millimeters
    *millimeters += d;

    // Adjust kilometers and centimeters if millimeters exceed their
respective units
    *centimeters += *millimeters / 1000;
```

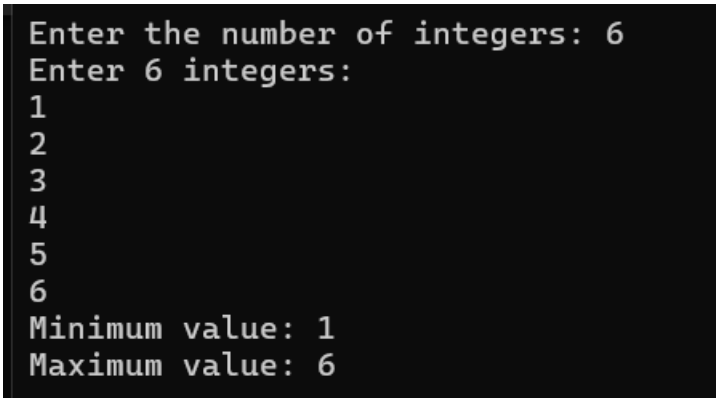
```
    *millimeters %= 1000;

    *kilometers += *centimeters / 100000;
    *centimeters %= 100000;
}

int main() {
    int kilometers, centimeters, millimeters, d;

    // Accept distance and d from the user
    printf("Enter the distance in kilometers, centimeters, and
millimeters (separated by spaces): ");
    scanf("%d %d %d", &kilometers, &centimeters, &millimeters);

    printf("Enter the number of millimeters to increment the distance
by: ");
    scanf("%d", &d);

    // Increment the distance using the function
    incrementDistance(&kilometers, &centimeters, &millimeters, d);

    // Display the new distance
    printf("New distance: %d kilometers, %d centimeters, %d
millimeters\n", kilometers, centimeters, millimeters);

    return 0;
}
```

```
Enter the distance in kilometers, centimeters, and millimeters (separated by spaces): 2
4
6
Enter the number of millimeters to increment the distance by: 2
New distance: 2 kilometers, 4 centimeters, 8 millimeters
```

4. Accept n integers in array A in main. Write a function which takes this array as parameter and find minimum and maximum from it. Display these values in main.

```
#include <stdio.h>

void findMinMax(int arr[], int n, int *min, int *max) {
```

```
    *min = *max = arr[0]; // Initialize min and max with the first
element of the array

    // Iterate through the array to find the minimum and maximum
elements
    for (int i = 1; i < n; i++) {
        if (arr[i] < *min) {
            *min = arr[i];
        }
        if (arr[i] > *max) {
            *max = arr[i];
        }
    }
}

int main() {
    int n;
    printf("Enter the number of integers: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d integers:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int min, max;
    findMinMax(arr, n, &min, &max);

    printf("Minimum value: %d\n", min);
    printf("Maximum value: %d\n", max);

    return 0;
}
```



Set C:

1. Accept date (dd, mm yy). Write a function to add no of days to the date and display the new date. Pass dd, mm and yy to the function using pointers.

```
#include <stdio.h>

void addDays(int *day, int *month, int *year, int daysToAdd) {
    // Array to store the number of days in each month
    int daysInMonth[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

    // Update the day by adding the number of days
    *day += daysToAdd;

    // Check if the year is a leap year
    if ((*year % 4 == 0 && *year % 100 != 0) || (*year % 400 == 0))
        daysInMonth[2] = 29; // Update February days for leap year

    // Adjust the day and month if the new day exceeds the days in the
month
    while (*day > daysInMonth[*month]) {
        *day -= daysInMonth[*month];
        (*month)++;
        if (*month > 12) {
            (*year)++;
            *month = 1;
        }
```

```
    }
}

int main() {
    int day, month, year, daysToAdd;

    // Input date
    printf("Enter date (dd mm yy): ");
    scanf("%d %d %d", &day, &month, &year);

    // Input number of days to add
    printf("Enter number of days to add: ");
    scanf("%d", &daysToAdd);

    // Call function to add days to the date
    addDays(&day, &month, &year, daysToAdd);

    // Display the new date
    printf("New date after adding %d days: %02d-%02d-%04d\n",
daysToAdd, day, month, year);

    return 0;
}
```

```
Enter date (dd mm yy): 01
12
2005
Enter number of days to add: 2
New date after adding 2 days: 03-12-2005
```

2. Write a function which accepts a number and three flags as parameters. If the number is even, setthefirstflagto1.Ifthenumberisprime,setthesecondflagto1.Ifthenumberisdivisibleby3 or 7, set the third flag to 1. In main, accept an integer and use this function to check if it is even, prime and divisible by 3 or 7.

```
#include <stdio.h>

// Function to check if a number is prime
int isPrime(int num) {
    if (num <= 1) return 0; // 1 and below are not prime
```

```
    if (num <= 3) return 1; // 2 and 3 are prime

    // Check if the number is divisible by any number from 2 to sqrt(num)
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) return 0; // Not prime if divisible
    }
    return 1; // Prime if not divisible by any number
}

// Function to set flags based on number properties
void setFlags(int num, int *flag1, int *flag2, int *flag3) {
    // Check if number is even
    if (num % 2 == 0) {
        *flag1 = 1;
    }

    // Check if number is prime
    if (isPrime(num)) {
        *flag2 = 1;
    }

    // Check if number is divisible by 3 or 7
    if (num % 3 == 0 || num % 7 == 0) {
        *flag3 = 1;
    }
}

int main() {
    int number, flag1 = 0, flag2 = 0, flag3 = 0;

    // Input an integer
    printf("Enter an integer: ");
    scanf("%d", &number);

    // Call function to set flags
    setFlags(number, &flag1, &flag2, &flag3);
```

```
    // Display flags
    printf("Number is even: %d\n", flag1);
    printf("Number is prime: %d\n", flag2);
    printf("Number is divisible by 3 or 7: %d\n", flag3);

    return 0;
}
```

```
Enter an integer: 24
Number is even: 1
Number is prime: 0
Number is divisible by 3 or 7: 1
```

Assignment 2: :Dynamicmemory allocation in C.

Set A:

1. The following program illustrate how to accept N integers and store them dynamically and access them.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
int *p, n,i;
 printf("How many elements:");
 scanf("%d",&n);
p = (int*)malloc(n*sizeof(int));
 for(i=0; i<n;i++)
 scanf("%d",p+i);
for(i=0; i<n;i++)
printf("%d\t",(p+i));
getch();
clrscr();
return 0;
}
```

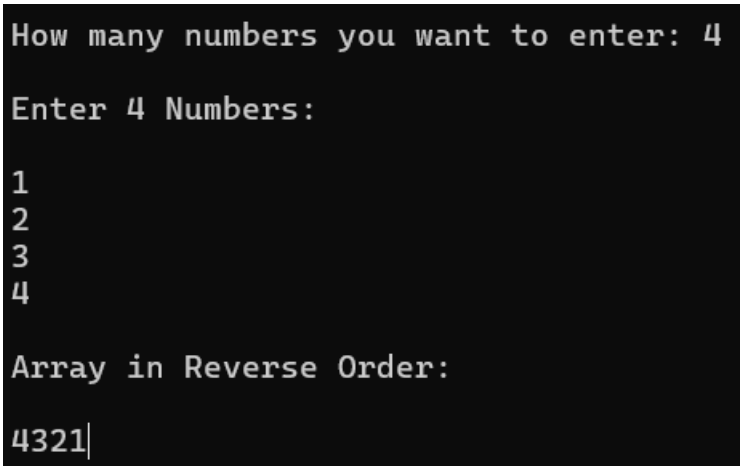
2. Type the above sample program 2 and execute it.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
 int *a[10],r,c,i,j;
printf("Enter the order of matrix\n");
scanf("%d%d",&r,&c);
printf("Enter matrix elements\n");
 for(i=0;i<r;i++)
 {
 /**** dynamically allocate memory for every row ****/
 a[i]=(int *)malloc(c*sizeof(int));
 for(j=0;j<c;j++)
 {
scanf("%d",a[i]+j);
 }
 }
 /****** Display Matrix ******/
printf("The matrix is as below\n");
 for(i=0;i<r;i++)
 {
 for(j=0;j<c;j++)
 {
printf("%d\t",*(*(a+i)+j));
 }
printf("\n");
 }
}
```

```
Enter the order of matrix
2
2
Enter matrix elements
1
2
3
4
The matrix is as below
1       2
3       4
```

3. Write a program to accept N integers and store them dynamically display them in reverse order.

```
#include<stdio.h>
#ibnlude<conio.h>
#include<stdlib.h>
int main()
{
int *p,n,i;
printf("How many numbers you want to enter: ");
 scanf("%d",&n);
p=(int*)malloc(n * sizeof(int));
printf("\nEnter %d Numbers:\n\n",n);
for(i=0;i<n;i++)
{
scanf("%d",p+i);
}
printf("\nArray in Reverse Order: \n\n");
for(i=n-1;i>=0;i--)
{
 printf("%d",*(p+i));
}
getch();
return 0;
}
```



4. Write a program to allocate memory dynamically for n integers such that the memory is initializedto0. And display it.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
int *p, n, i;
printf("enter the number of elements:");
scanf("%d",&n);
p=(int*)calloc(n,sizeof(int));
if(p==NULL)
{
        printf("memory are not allocated");
        exit(0);
}
else
{
        for(i=0;i<n;i++)
        {
                scanf("%d",p+i);

        }
}

for(i=0; i<n;i++)
```

```
        printf("%d\t",*(p+i));

getch();
return 0;
}
```

```
enter the number of elements:4
1
2
3
4
1       2       3       4
```

Set B:

1. Accept N integers in an array using dynamic memory allocation. Find maximum from them and display.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
int *p, n, i, max;
printf("enter the n number:");
scanf("%d", &n);
p=(int*)malloc(n*sizeof(int));
printf("enter the elements:");
for(i=0; i<n; i++)
{
scanf("%d", &p[i]);
printf("\n");
}
max=p[0];
for(i=0;i<n;i++)
{
if(max<p[i])
{
max=p[i];
}
}
```

```
printf("maximum number is: %d\n", max);
getch();
return 0;
}
```

```
enter the n number:4
enter the elements:1

2

3

4

maximum number is: 4
```

2. Accept n integers in an array. Copy only the non-zero elements to another array (allocated using

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
int main(){
int a[30], i,n, sum = 0;
int count=0;
int *ptr;
float avg;
printf("\n Enter the total number of elements you want to enter : ");
scanf("%d",&n);
printf("\n Enter element in aaray:");
for(i = 0;i<n;i++){
scanf("%d",&a[i]); }
for(i = 0;i<n;i++){
if(a[i]!=0)
{
count++;
 }
}
ptr = (int *)malloc(count * sizeof(int));
for(i = 0;i<n;i++)
```

```
{
if(a[i]!=0)
{
*(ptr+i)=a[i];
sum=sum+*(ptr+i);
}
}
avg=sum/count;
printf("sum is %d \n",sum);
printf("avg is %f\n",avg);
}
```

```
 Enter the total number of elements you want to enter : 5

 Enter element in aaray:1
2
3
4
5
sum is 15
avg is 3.000000
```

Assignment 3: String handling in C and standard library functions for strings

Set A :

1. Write a program to read a string and copy it to another sting and display copied string. Also the length of copied string. (Use built in functions)

```
#include <stdio.h>
#include<conio.h>
#include <string.h>
int main()
 {
 char str1[20],str2[20];
printf("enter the string:");
gets(str1);
```

```
printf("\n string 1:%s",str1);
strcpy(str2,str1);
printf(" \n string 2:");
printf("\n length is:%d",strlen(str1));
getch();
return 0;
}
```

```
enter the string:silviya

 string 1:silviya
 string 2:
 length is:7
```

2. Write a program to read two strings. If first string is greater than second then concatenate second to first and display concatenated string, If first string is smaller than second then concatenate first to second, other-wise display length of string. (use strcmp)

```
#include <stdio.h>
#include<conio.h>
#include <string.h>
int main()
 {
char str1[1000], str2[1000];
int comparison_result;
printf("Enter the first string: ");
gets(str1);
printf("Enter the second string: ");
 gets(str2);
strcmp(str1, str2);
if (comparison_result == 0) {
strcat(str1, str2);
 printf("Concatenated string: %s\n", str1);
}
else if (comparison_result < 0)
{
strcat(str2, str1);
printf("Concatenated string: %s\n", str2);
}
 else
```

```
{
printf("Length of the first string: %lu\n", strlen(str1));
getch();
return 0;
}
}
```

```
Enter the first string: joy
Enter the second string: mark
Concatenated string: joymark
```

3. Write a program to read a string and one character. Check whether given character is present in the given string or not? (Hint: use strchr or strrchr)

```
#include <stdio.h>
#include <string.h>
int main()
{
 char a, word[50];
 int i, freq = 0, flag = 0;
 printf("Enter character: ");
 scanf("%c", &a);
 printf("Now enter the word:\n ");
 scanf("%s", word);
 printf("Positions of '%c' in %s are: ", a, word);
 for (i = 0; i < strlen(word); i++)
 {
 if (word[i] == a)
 {
 flag = 1;
 printf("%d  ", i + 1);
 freq++;
 }
 }
 if (flag)
 {
 printf("\nCharacter '%c' occured for %d times.\n", a, freq);
 }
 else
```

```
  {
  printf("None\n");
   }
return 0;
}
```

```
Enter character: piya
Now enter the word:
 Positions of 'p' in iya are: None
```

Set B:

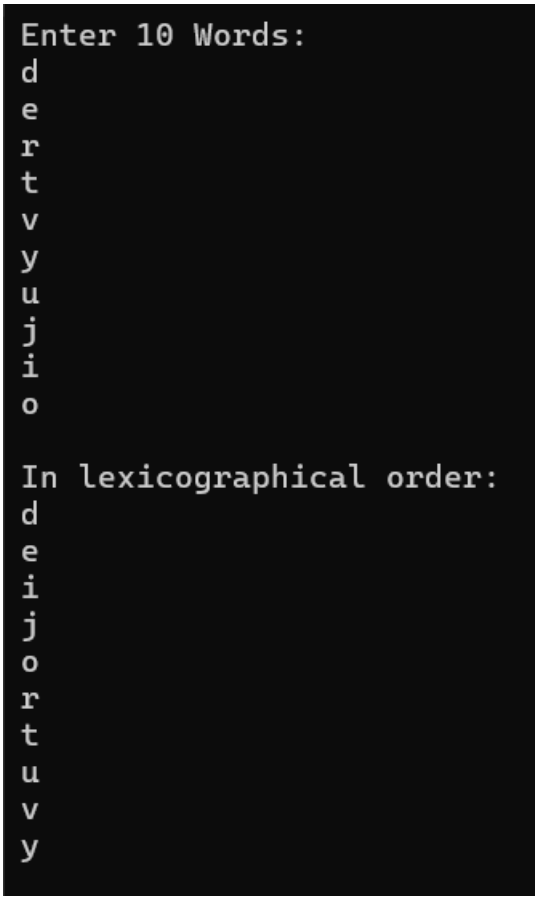
2.Write a program which accepts a sentence from the user. Find and display reverse of it. (Don’t use any function)

```
#include<stdio.h>
#include<string.h>
int main()
{
int c = 0;
char ch, s[100];
printf("Input a string:\n");
gets(s);
for(c=0;s[c] != '\0';c++) {
ch = s[c];
if (ch >= 'A' && ch <= 'Z')
s[c] = s[c] + 32;
else if (ch >= 'a' && ch <= 'z')
s[c] = s[c] - 32;
else if(s[c]==' ')
s[c] = '*';
else if(s[c]>0 || s[c]<9)
s[c] = '?';
}
puts(s);
return 0;
}
```

```
Input a string:
riya shinde 80
RIYA*SHINDE*??
```

3. Write a program that accepts n words and outputs them in dictionary order.

```
#include<stdio.h>
#include<string.h>
int main()
{
  char str[10][50],temp[50];
  int i,j;

  printf("Enter 10 Words:\n");
  for(i=0;i<10;i++)
  scanf("%s[^\n]",str[i]);
   for(i=0;i<9;i++)
  {
  for(j=i+1;j<10;j++)
 {
 if(strcmp(str[i],str[j])>0)
 {
 strcpy(temp,str[i]);
 strcpy(str[i],str[j]);
 strcpy(str[j],temp);
 }
 }
 }
 printf("\nIn lexicographical order: \n");
for(i=0;i<10;i++)
puts(str[i]);
return 0;
}
```



Assignment 4: String handling using user defined function and pointers.

Set A:

1. Write a program to accept a string and find its length using user defined function. (Don't use pointers)

```
#include <stdio.h>
#include<conio.h>
#include <string.h>
int main()
 {
 char str1[20];
printf("enter the string:");
gets(str1);
printf("\n string 1:%s",str1);
printf("\n string length is:%d",strlen(str1));
```

```
getch();
return 0;
}
```

```
enter the string:sahil

 string 1:sahil
 string length is:5
```

2. Write a function that takes a string as parameterand returns the same string in upper case(use pointes). Accept this string in main and display converted string in main only.

```
#include <stdio.h>
#include<conio.h>
#include <string.h>
int main()
{
char s[100];
int i;
printf("Enter a string : ");
gets(s);
for (i = 0; s[i]!='\0'; i++)
 {
 if(s[i] >= 'a' && s[i] <= 'z')
 {
s [i] = s[i] -32;
  }
  }
 printf("String in Upper Case = %s", s);
 getch();
return 0;
}
```

```
Enter a string : ram
String in Upper Case = RAM
```

3. Write a function to find reverse of the string and use it in main.

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

```
#include<string.h>
int main()
{
int n,i;
char str[20];
printf("enter the string to get reversed:");
gets(str);
n=strlen(str);
printf("\n reversed string is\n:");
for(i=n-1;i>=0;i--)
{
printf("%c",str[i]);
}
getch();
return 0;
}
```

```
enter the string to get reversed:sam

 reversed string is
:mas
```

Set B:

1. Write a function to compare two strings. Write another function to reverse the string. In main function a string and check whether it is palindrome or not using above functions. (Hint: A palindrome string is a string which reads same in forward as well as backward direction for example: madam, nitin, etc.)

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
bool compareStrings(const char *str1, const char *str2) {
   return strcmp(str1, str2) == 0;
}
void reverseString(char *str) {
   int length = strlen(str);
   int i, j;
   char temp;
```

```
    for (i = 0, j = length - 1; i < j; i++, j--) {
        temp = str[i];
        str[i] = str[j];
        str[j] = temp;
    }
}

int main() {
    char str[100];

    printf("Enter a string: ");
    scanf("%s", str);

    char original[100];
    strcpy(original, str);

    reverseString(str);
    if (compareStrings(original, str)) {
        printf("The string is a palindrome.\n");
    } else {
        printf("The string is not a palindrome.\n");
    }

    return 0;
}
```

```
Enter a string: jay
The string is not a palindrome.
```

```
Enter a string: jay
The string is not a palindrome.
```

Assignment 5: Structures and union in C

Set A :

1. Create a structure employee (id, name, salary). Accept details of n employees and display display it in summary format.

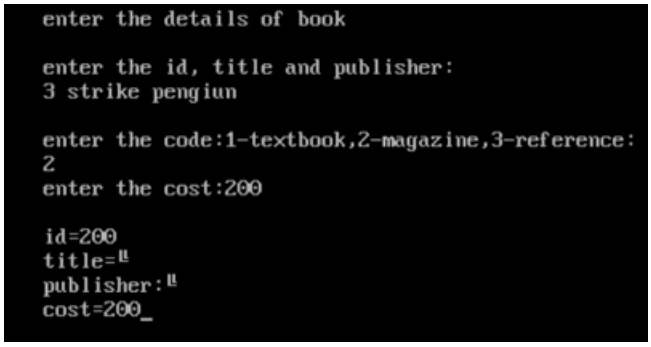
```
#include <stdio.h>
#include<conio.h>
struct employee
{
char    name[30];
int     empId;
float   salary;
};
int main()
{
struct employee emp;
printf("\nEnter details :\n");
printf("Name :");
gets(emp.name);
printf("ID :");
scanf("%d",&emp.empId);
printf("Salary :");
scanf("%f",&emp.salary);
printf("\nEntered detail is:\n");
printf("Name: %s\n",emp.name);
printf("Id: %d \n",emp.empId);
printf("Salary:%f\n",emp.salary);
getch();
return 0;
}
```

```
Enter details :
Name :riya
ID :201
Salary :45000

Entered detail is:
Name: riya
Id: 201
Salary:45000.000000
```

2.Type sample program 2 and execute it.

```
#include<stdio.h>
#include<coino,h>
struct library book
int id:
char title[80],publisher[20];
int code;
 union u
{
int no_of_copies;
char month[10];
 int edition;
}
 info;
intcost,
};
void main()
{
struct library_book book 1;
printf("\n Enter the details of the book \n");
printf("\n Enter the id, title and publisher \n");
scanf("%d%s%s", &book1.id, book1.title, book1.publisher);
 printf("\nEnterthecode: 1-TextBook, 2-Magazine, 3-Reference");
 scanf("%d",book I.code);
switch(book1.code)
case 1: printf("Enter the number of copies:");
scanf("%d",&book1.info.no of copies);
break;
```

```
 case2:
printf("Enter the issue month name:");
 scanf("%s", book1.info.month);
break;
 case3:
 printf("Enter the editionnumber:");
scanf("%d",&book1.info.edition);
break;
}
printf("Enter the cost:");
scanf("%d",&book1.cost);
printf("\n id %d", book 1.id);
printf("n Title %s", book1.title);
printf("\n Publisher %s", book1.publisher);
switch(book1.code)
{
case 1:printf("Copies %d:",book1.info.no of copies);
break;
case2: printf("Issue month name=%s", book1.info.month);
break;
case3:printf("Edition number=%d:",book1.info.edition);
break
}
printf("\n Cost-%d", book1.cost);
getch();
clrscr();
return 0;
}
```



Set B :

2.Create a structure car (car number, model name, colour, cost). Accept details of n cars and write amenu driven program to perform the following operations. Write separate functions for the different options. i) Search byname ii) Displayal.

include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

struct node{

char model[10],color[10];

int year;

struct node *next;

};

struct node *temp,*head;

void display(struct node *head){

temp=head;

while(temp!=NULL){

if(temp->year>2010 && (strcmp("yellow",temp->color)==0))

printf(" %s \t\t %s \t\t %d",temp->model,temp->color,temp->year);

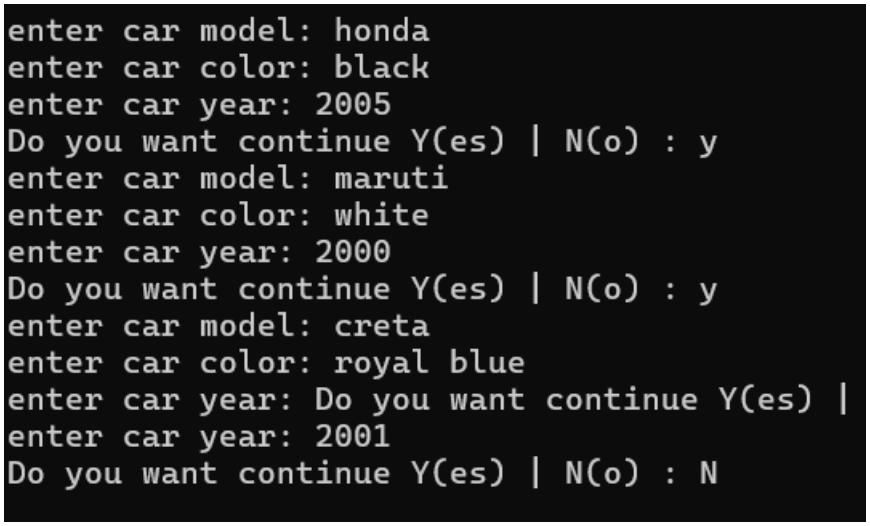
temp=temp->next;

printf("");

}

}

```
int main(){
int n;
char option,enter;
head=(struct node *)malloc(sizeof(struct node));
temp=head;
do{
printf("enter car model: ");
scanf("%s",temp->model);
printf("enter car color: ");
scanf("%s",temp->color);
printf("enter car year: ");
scanf("%d",&temp->year);
printf("Do you want continue Y(es) | N(o) : ");
scanf("%c",&enter);
scanf("%c",&option);
if (option!='N'){
temp->next=(struct node *)malloc(sizeof(struct node));
temp=temp->next;
}
else
 {
temp->next=NULL;
}
}while(option!='N');
display(head);
getch();
return 0;
}
```



3.Create a structure movie (name, release year, duration). Accept details of n movies. Write a function to sort them according to release year. Display them in main() function. (Hint: Use dynamic memoryallocation.)

Assignment 6: File Handling (Text Files)

Set A :

1. Write a program to accept a file name and a single character. Display the count indicating total number of times character occurs in thefile

```
#include <stdio.h>
int main() {
    FILE *file;
    char filename[100], character;
    int count = 0;

    printf("Enter the file name: ");
    scanf("%s", filename);
    printf("Enter the character to count: ");
    scanf(" %c", &character);

    file = fopen(filename, "r");
    if (file == NULL) {
        printf("Error opening file.\n");
        return 1;
    }

    char ch;
    while ((ch = fgetc(file)) != EOF) {
        if (ch == character) {
            count++;
        }
    }

    fclose(file);

    printf("The character '%c' occurs %d times in the file '%s'.\n",
character, count, filename);

    return 0;
}
```

```
Enter the file name: A6setA1.cpp
Enter the character to count: A
The character 'A' occurs 0 times in the file 'A6setA1.cpp'.
```

2.Write a program to accept two filenames. Copy the contents of the first file to the second such that the case of all alphabets isreversed.

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

int main() {
    FILE *source_file, *destination_file;
    char source_filename[100], destination_filename[100];
    char ch;

    printf("Enter the name of the source file: ");
    scanf("%s", source_filename);
    printf("Enter the name of the destination file: ");
    scanf("%s", destination_filename);

    source_file = fopen(source_filename, "r");
    if (source_file == NULL) {
        printf("Error opening source file.\n");
        return 1;
    }

    destination_file = fopen(destination_filename, "w");
    if (destination_file == NULL) {
```

```
        printf("Error opening destination file.\n");
        fclose(source_file);
        return 1;
    }

    while ((ch = fgetc(source_file)) != EOF) {
        if (isalpha(ch)) {
            if (islower(ch)) {
                ch = toupper(ch);
            } else {
                ch = tolower(ch);
            }
        }
        fputc(ch, destination_file);
    }

    fclose(source_file);
    fclose(destination_file);

    printf("Contents of '%s' copied to '%s' with reversed case.\n",
source_filename, destination_filename);

    return 0;
}
```

```
Enter the name of the source file: A6setA2.cpp
Enter the name of the destination file:  A6setA2.txt
Contents of 'A6setA2.cpp' copied to 'A6setA2.txt' with reversed case.
```

3.Write a program to accept a filename and count the number of words, lines and characters in thefile.

```
#include <stdio.h>
int main() {
   FILE *file;
   char filename[100];
   char ch;
   int word_count = 0, line_count = 0, char_count = 0;

   printf("Enter the file name: ");
   scanf("%s", filename);

   file = fopen(filename, "r");
   if (file == NULL) {
      printf("Error opening file.\n");
      return 1;
   }

   while ((ch = fgetc(file)) != EOF) {
      char_count++;

      if (ch == ' ' || ch == '\t' || ch == '\n') {
         word_count++;
      }
```

```
        if (ch == '\n') {
            line_count++;
        }
    }

    if (char_count > 0) {
        word_count++;
    }

    if (char_count > 0) {
        line_count++;
    }

    fclose(file);

    printf("Number of words in the file: %d\n", word_count);
    printf("Number of lines in the file: %d\n", line_count);
    printf("Number of characters in the file: %d\n", char_count);

    return 0;
}
```

```
Enter the file name: A6setA3.cpp
Number of words in the file: 304
Number of lines in the file: 45
Number of characters in the file: 886
```

Set B :

2. Write a program to read integers from a file and write even and odd numbers in separate files.

```
#include <stdio.h>
int main() {
   FILE *file;
   char filename[100];
   char ch;
   int word_count = 0, line_count = 0, char_count = 0;

   printf("Enter the file name: ");
   scanf("%s", filename);

   file = fopen(filename, "r");
   if (file == NULL) {
      printf("Error opening file.\n");
      return 1;
   }

   while ((ch = fgetc(file)) != EOF) {
      char_count++;

      if (ch == ' ' || ch == '\t' || ch == '\n') {
         word_count++;
      }

      if (ch == '\n') {
         line_count++;
      }
   }

   if (char_count > 0) {
      word_count++;
   }

   if (char_count > 0) {
      line_count++;
```

```
    }

    fclose(file);

    printf("Number of words in the file: %d\n", word_count);
    printf("Number of lines in the file: %d\n", line_count);
    printf("Number of characters in the file: %d\n", char_count);

    return 0;
}
```

```
Enter the file name: A6setB2.cpp
Number of words in the file: 349
Number of lines in the file: 50
Number of characters in the file: 1244
```

3. Write a program to compare two files character by character and check whether they are same or not?

```
#include <stdio.h>
int main() {
    FILE *file1, *file2;
    char filename1[100], filename2[100];
    char ch1, ch2;
    int different = 0;

    printf("Enter the name of the first file: ");
    scanf("%s", filename1);
    printf("Enter the name of the second file: ");
    scanf("%s", filename2);

    file1 = fopen(filename1, "r");
    if (file1 == NULL) {
        printf("Error opening first file.\n");
        return 1;
    }

    file2 = fopen(filename2, "r");
    if (file2 == NULL) {
```

```
        printf("Error opening second file.\n");
        fclose(file1);
        return 1;
    }

    while ((ch1 = fgetc(file1)) != EOF && (ch2 = fgetc(file2)) != EOF) {
        if (ch1 != ch2) {
            different = 1;
            break;
        }
    }

    if (ch1 != EOF || ch2 != EOF) {
        different = 1;
    }

    fclose(file1);
    fclose(file2);

    if (different) {
        printf("Files are different.\n");
    } else {
        printf("Files are the same.\n");
    }

    return 0;
}
```

```
Enter the name of the first file: A6setB3.cpp
Enter the name of the second file: A6setB2.cpp
Files are different.
```

Assignment 7: Command line arguments and preprocessor directives.

Set A :

1.Write a program to define value of pi and use it to find area and perimeter of a circle.

```
#include <stdio.h>
#define PI 3.14159
int main() {
    float radius, area, perimeter;

    printf("Enter the radius of the circle: ");
    scanf("%f", &radius);

    area = PI * radius * radius;
    perimeter = 2 * PI * radius;

    printf("Area of the circle: %.2f\n", area);
    printf("Perimeter of the circle: %.2f\n", perimeter);

    return 0;
}
```

```
Enter the radius of the circle: 24
Area of the circle: 1809.56
Perimeter of the circle: 150.80
```

.

2. Define a macro MIN which gives the minimum of two numbers. Use this macro to find the minimum of nnumbers.

```
#include <stdio.h>
#define MIN(x, y) (((x) < (y)) ? (x) : (y))

int findMinimum(int arr[], int n) {
    int min = arr[0];
    for (int i = 1; i < n; i++) {
        min = MIN(min, arr[i]);
    }
    return min;
}
int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d numbers: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int min = findMinimum(arr, n);
    printf("Minimum number: %d\n", min);

    return 0;
}
```

```
Enter the number of elements: 4
Enter 4 numbers: 1
2
3
4
Minimum number: 1
```

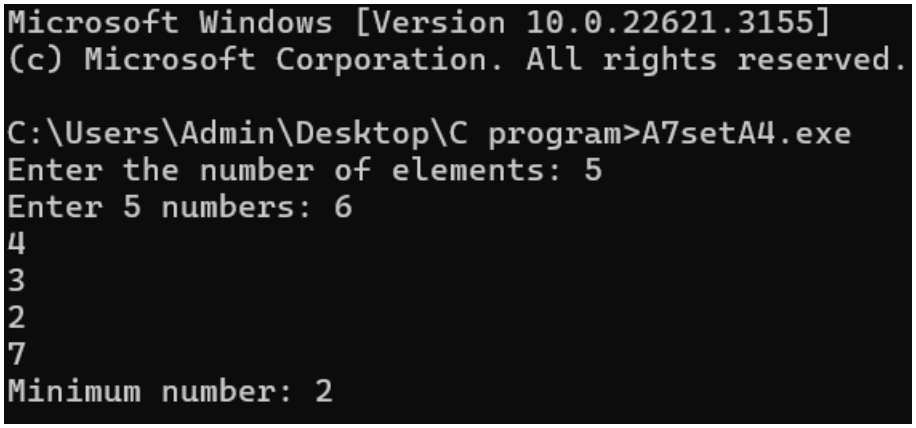
3.Write a program to display all command line arguments passed to main in the reverse order.

```
#include <stdio.h>

int main(int argc,char* argv[])

{

	for(int i=0; i<argc; i++)

	{

		printf("%s\n",argv[i]);

	}

}
```

```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\C program>A7setA2.exe
A7setA2.exe

C:\Users\Admin\Desktop\C program>A7setA2.exe 1234
A7setA2.exe
1234

C:\Users\Admin\Desktop\C program>A7setA2.exe hi im rishab
A7setA2.exe
hi
im
rishab
```

4.Write a program to accept three integers as command line arguments and find the minimum, maximum and average of the three. Display error message if invalid number of arguments are entered.

```
#include <stdio.h>

#define MIN(x, y) (((x) < (y)) ? (x) : (y))

int findMinimum(int arr[], int n) {
    int min = arr[0];
    for (int i = 1; i < n; i++) {
        min = MIN(min, arr[i]);
    }
    return min;
}

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d numbers: ", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int min = findMinimum(arr, n);
    printf("Minimum number: %d\n", min);

    return 0;
}
```



Set B :

1.Write a program to accept two file names as command line argument and copy the contents of first file to second file.

```
#include <stdio.h>

int main(int argc, char *argv[]) {
   FILE *source_file, *destination_file;
   char ch;

   if (argc != 3) {
      printf("Usage: %s <source_file> <destination_file>\n", argv[0]);
      return 1;
   }

   source_file = fopen(argv[1], "r");
   if (source_file == NULL) {
      printf("Error: Unable to open source file.\n");
      return 1;
   }
```

```
    destination_file = fopen(argv[2], "w");
    if (destination_file == NULL) {
        printf("Error: Unable to open destination file.\n");
        fclose(source_file);
        return 1;
    }

    while ((ch = fgetc(source_file)) != EOF) {
        fputc(ch, destination_file);
    }

    fclose(source_file);
    fclose(destination_file);

    printf("Contents of '%s' copied to '%s' successfully.\n", argv[1], argv[2]);

    return 0;
}
```

```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\C program>A7setB3.exe
Usage: A7setB3.exe <source_file> <destination_file>

C:\Users\Admin\Desktop\C program>A7setA2.exe
A7setA2.exe

C:\Users\Admin\Desktop\C program>A7setA1.exe
Enter the radius of the circle: 24
Area of the circle: 1809.56
Perimeter of the circle: 150.80
```