* Name :- Vardhan Shrenikkumar Shah

* Roll No. :- 25BCB059

* Subject :- C language (IA 2)

* Topic 1 :- Simple programs

(1) Hours into minute -

```
#include <stdio.h>
int main() {
int hours;
printf("please enter hours:");
scanf("%d", &hours);
int minutes = hours * 60;
printf("%d hours is %d minutes\n",
        hours, minutes);
return 0;
}
```

(2) celsius into farenhit -

```
#include <stdio.h>
int main() {
float celsius, fahrenhit;
printf("Enter °C ");
scanf("%f", &celsius);
fahrenhit = (celsius * 9 / 5) + 32;
printf("%.2f celsius is equal to %.2f
        Fahrenhit\n", celsius, fahrenhit);
```

```
    return 0;
}
```

(3) Calculate area of a circle.

```
#include <stdio.h>
int main () {
    float radius, area;
    const float PI = 3.14159;
    printf ("Enter the radius of the circle: ");
    scanf ("%f", &radius);
    area = PI * radius * radius;
    printf (" The area of the circle with
radius %.2f is : %.2f \n", radius, area);
    return 0;
}
```

(4) calculate area of a triangle.

```
#include <stdio.h>
int main () {
    Float base, height, area;
printf (" Enter the base of the triangle :");
scanf ("%f", &base);
printf (" Enter the height of the triangle:"
); scanf (" %f", &height);
area = (base * height) /2 ;
printf (" The area of the triangle is : %.2f
\n", area);
return 0;
}
```

(5) Swap two values.

```
#include <stdio.h>
int main() {
    int a = 5, b = 10;
    int temp;
    printf("Before swapping: a = %d, b = %d\n",
            a, b);
    temp = a;
    a = b;
    b = temp;
    printf("After swapping: a = %d, b = %d\n
    ", a, b);
    return 0;
}
```

* Topic 2 :- Using if condition

(1) largest and smallest of two values.

```
#include <stdio.h>
int main() {
    int a, b;
    printf("write two values:");
    scanf("%d %d", &a, &b);
    if (a > b) b is smallest
    printf("a is largest");
else
    printf("a is smallest and b is
    largest");
    return 0;
}
```

(2) largest and smallest of three values.

```
#include <stdio.h>
int main () {
int a, b, c ;
Printf("Write three values");
Scanf("%d %d %d", &a, &b, &c);
if ( a≥b && a ≥ c) {
  printf(" a is largest"); }
else if ( b ≥a && b≥c) {
  printf(" b is largest"); }
else
{printf(" c is largest"); };
if ( a≤b && a≤c) { printf("a is
 smallest "); }
else if ( b ≤ c && b≤a) { printf("b i
 smallest "); }
else
      printf(" c is smallest"); }
return 0; }
```

(3) Find out net salary.

```
#include <stdio.h>
int main () {
int gsalary, net salary, allowances, dedu
printf(" write gross salary");
Scanf(" %d", &gsalary);
if ( gsalary > 10000) {net salary
```

= gsalary + 0.1 x gsalary - 0.03 x gsalary; }

else if (gsalary > 5000) {

netsalary = gsalary + 0.07 gsalary - 0.02 gsalary; }

printf ("%d", netsalary);
return 0;
}

(4) find number is divisible by 7 or not:

```
#include <stdio.h>
int main () {
int a;
printf (" write a number");
scanf (" %d", &a);
if ( a%7 ==0) {
printf (" number is divisible by 7");
else if ( a%7 != 0) {
printf (" number is not divisible by 7");
return 0; }
```

(5) find out net sales.

```
#include <stdio.h>
int main () {
int grosssale, netsale;
printf (" write grosssale ");
Scanf (" %d", &grosssale);
```

```
if ( grosssale>20000 ) {
  printf ("net sale is %d", grosssale -
                              0.15*grosssale
    ;) ; }
else if ( grosssale > 10000) {

printf (" net sale is %d", grosssale -
          0.05 * grosssale ); };
  return 0;
  }
```

* Topic 3 :- Using loop.

(1) print sum of 1st n. odd nos.

```
#include <stdio.h>
int main () {
int i,n; int sum;
printf ("write a number");
scanf ("%d", &n);
for (i=1; i<n ; i++ ) {
    sum += (2*i-1); } }

printf ("%d", sum);
return 0; }
```

(2) print factorial of given number.

```
#include <stdio.h>
int main () {
int i,n; int factorial=1;
printf ("write a number");
scanf ("%d", &n);
for( i=n; i>=1; i--) {
    factorial *= i; }
printf ("%d", factorial);
return 0; }
```

(3) check a number is palindrome or not.
e.g :- 12321

```
#include <stdio.h>
int main () {
int num, reversed = 0, remainder,
original;
printf("Enter an integer : ");
scanf ("%d", &num);
while (num != 0) {
remainder = num % 10;
reversed = reversed * 10 + remainder;
num /= 10; }
if ( original == reversed)
 printf ("given number is palindrome");
else
printf(" given number is not palindrome
        ");
return 0; }
```

(4) find a given number is armstrong numb

```
#include <stdio.h> #include <math.
int main(){
int num, original, remainder, n=0;
float result = 0.0;
printf("Enter an integer:");
scanf("%d", &num);
original = num;
while(original != 0){
        original /= 10;
        ++n; }
original = num;
while(original != 0){
 remainder = original %10;
 result += pow(remainder, n);
 original /= 10; }

if(result == num)
 printf("yes");
else printf("NO");
return 0; }
```

(5) find out given number is perfect or not.

```
#include <stdio.h>
int main(){
int num, sum = 0;
printf("Enter positive integer");
scanf("%d", &num);
```

```
for(int i = 1; i<num; i++) {
    if (num % i == 0)
        sum += i;
    }

if(sum == num)
printf("It is perfect number");
    else printf("It is not perfect
     number !!");
     return 0; }
```

* Topic 4 :- ARRAYS

(1) Accept 5 values and sort the array.

```
#include <stdio.h>
int main() {
int arr[5], i, j, temp;
printf("Enter 5 integers");
For (i=0; i<5; i++) {
   Scanf("%d", &arr[i]); }
for(i=0; i<4; i++) {
for(j=0; j<4-i; j++) {
if(arr[j] > arr[j+1]) {
    temp = arr[j];
    arr[j] = arr[j+1];
    arr[j+1] = temp; } } }
printf("sorted array:");
for(i=0; i<5; i++) {
    printf("%d", arr[i]); }
    return 0; }
```

(2) Add two 2D array of same size and st the result in the 3rd one.

```
#include <stdio.h>
int main () {
 int r, c;
int mat1 [10] [10], mat2 [10][10],
   Sum [10] [10];
   scanf ("%d %d", &r, &C);

// take input for 2D array

for (int i=0; i<r; ++i) {
  for (int j=0; j<c; ++j) {
     Sum [i][j]= mat1 [i][j]
     + mat2 [i] [j]; } }
  printf ("%d", Sum [i][j]);
  return 0; }
```

(3) multiply two 2D array.

```
#include <stdio.h>
int main () {
  int A [2][2], B [2] [2], C [2] [2];
   int i, j, k;

// take input for matrix A and B.

  for (i= 0 ; i<2 ; i++) {
    for ( j=0; j<2; j++) {
       C [i] [j] = 0;
```

```
for (k=0; k<2; k++) {
    c[i][j] += A[i][k] * B[k][j];
} } }

printf("Result:");
for (i=0; i<2; i++) {
  for (j=0; j<2; j++) {
    printf("%d", c[i][j]); }
    printf("\n"); }
    return 0; }
```

(4) Reverse an array :-

```
#include <stdio.h>
int main() {
int arr[5], i, temp;
printf("Enter 5 integers:\n");
for (i=0; i<5; i++) {
  scanf("%d", &arr[i]);
for (i=0; i<5/2; i++) {
  temp = arr[i];
  arr[i] = arr[4-i];
  arr[4-i] = temp;
printf("Reversed array:\n");
for (i=0; i<5; i++) {
  printf("%d", arr[i]); }
  return 0; }
```

(5) delete a value from the array. (2)

```
#include <stdio.h>
int main (){
int arr[5], i, pos; size = 5;
printf("Enter 5 integers: \n");
for (i = 0; i < size; i++){
 scanf("%d", &arr[i]);
printf("enter position to delete");
scanf("%d", &pos);
for (i = pos; i < size-1; i++){
   arr[i] = arr[i+1];
 } size--;
printf("array after deletion: \n");
for(i=0; i<size; i++){
  printf("%d", arr[i]); }
  return 0; }
```

* Topic 5 :- STRINGS

(1) Convert a string to a lower case.

```
#include <stdio.h>
#include <string.h>
int main (){
  char str[] = "Hi";
  printf("%s\n", strlwr(str));
  return 0; }
```

(2) convert a string to toggle case.

```
#include <stdio.h>
# include <string.h>
void togglecase(char *str) {
 for (int i = 0; str[i] != '\0'; i++) {
 if (str[i] >= 'a' && str[i] <= 'z') {
   str[i] = str[i] - 32; }
 else if (str[i] >= 'A' && str[i] <=
  'z' ) {  str[i] = str[i] + 32; }
   } }
  int main() {
    char mystring[] = "Hello World";
    toggle case (mystring);
    printf ("Toggled string: %s",
     mystring); return 0; }
```

(3) concentrate one string to end of second string.

```
#include <stdio.h>
# include <string.h>
int main () {
 char str1[] = "Hello, ";
 char str2[] = "World";
 strcat (str1, str2);
printf("concatenated string: %s",
 str1);
 return 0; }
```

Output :- HelloWorld

(4) check whether a string is palindrom or not.

```
#include <stdio.h>
#include <string.h>
int main(){ printf("enter a string:");
    char str[100];
    scanf("%s", str);
    if(strcmp(str == strrev(str)))
        { printf("It is palindrome");
    else
        printf("It is not palindrome");
    return 0;}
```

(5) print frequency of each vowel in a given string.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int main() { char str[100];
        int a=0, e=0, i=0, o=0, u=0;
printf("enter a string:");
 gets(str, sizeof(str), stdin);
 for(int j=0; j<strlen(str); j++){
 char ch=tolower(str[j]);
 if (ch=='a') a++;
 .... else if (ch=='u') u++;
  printf("vowel frequencies:");
  printf("a, e, i, o, u: %d, %d, %d, %d, %d");
   return 0;}
```

* Topic 6 :- Pointers, Arrays, strings

1. addition, subtraction and multiplication of two matrices.

```
#include <stdio.h>
int main() {
int arr[2][2] = { { 1, 2 },
                  { 3, 4 } };
int arr1[2][2] = { { 3, 4 },
                   { 5, 6 } };
for (int i=0; i<2; i++) {
    for (int j=0; j<2; j++) {
arr2[i][j] = arr[i][j] + arr1[i][j];
arr2[i][j] = arr[i][j] - arr1[i][j];
for (int k=0; k<2; k++) {
arr2[i][j] += arr[i][k] * arr1[k][j];
    } } }

for (int i=0; i<2; i++) {
    for (int j=0; j<2; j++) {
    printf("%d\n", arr2[i][j]); } }
return 0; }
```

2. Sort all the elements of a 4x4 matrix and store the result in a single-dimension array.

```
#include <stdio.h>
int main() {
int matrix[ ][ ];
int array[ ];
```

```
int k = 0 ;
Printf (" Enter elements Of a 4x4
matrix : \n ");
 for ( int i = 0; i < size; i++ ) {
    for ( int j = 0; j < size; j++ ) {
  scanf ( "%od ", &matrix [i] [j] );
  array [k++] = matrix [i] [j] ;
    } }
 for ( int i=0; i < Total - 1; i++ ) {
  for ( int j = 0; j < Total - i - 1; j++ ) {
    if ( array [j] > array [j + 1] ) {
    int temp = array [j] ;
    array [j] = array [j + 1] ;
    array [j+1] = temp ;
    } } }
  for ( int i=0; i < Total; i++ ) {
  printf ( "%od ", array [i] ) ; }
  return 0; }
```

(3) Print the largest and smallest numbers From a 3x3 matrix using Pointer in C

```
# include <stdio.h>  (size=3)
int main () {
int matrix [size] [size];
int *ptr = &matrix [0] [0];
int max, min ;
printf (" enter elements : ") ;
for ( int i = 0; i < SIZE ; i++ )
  for ( int j = 0 ; j < SIZE ; j++ )
    scanf ( "%od ", &matrix [i] [j]);
```

```
max = min = *ptr;

for (int i = 0; i < SIZE * SIZE; i++) {
  if (*(ptr + i) > max)
      max = *(ptr + i);
  if (*(ptr + i) < min)
      min = *(ptr + i); }

printf("Largest element: ", max);
printf("Smallest element: ", min);
return 0;
}
```

(4) Accept and print 'later on three books names using array of pointers.

```
#include <stdio.h>
int main() {
  char *books[3];
  char temp[100];
printf("Enter names of 3 books:");
for (int i = 0; i < 3; i++) {
 printf("BOOK %d:", i+1);
 fgets(temp, sizeof(temp) stdin);
 books[i] = malloc(strlen(temp)+1);
 strcpy(books[i], temp); }
 for (int i=0; i<3; i++) {
 printf("BOOK %d: %s", i+1, books[i]);
 free(books[i]); }
 return 0; }
```

(5) Write a program that takes a set of names of individuals and abbreviates the first, middle and other names except the last name by their first letter.

```
#include <stdio.h>
#include <string.h>
void abbreviateName (char *fullName) {
    char *token;
    char nameparts[10][50];
    int count = 0;
    token = strtok (fullName, "n");
    while (token != Null) {
    strcpy (nameparts [count++], token);
    token = strtok (Null, "\n");
    for (int i=0; i < count-1; i++) {
    printf ("%0c.", nameparts [i][0]);
} printf ("%0s\n", nameparts
    [count-1]);
int main () {
    int n;
    char fullName [200];
    scanf ("%0d", &n);
    getchar ();
    for ( int i=0; i<n; i++) {
    fgets ( fullName, sizeof (full
    Name), stdin);
    abbreviatename (fullname);
} return 0;}
```

Calling function

Output :- John Joy Wich (Input)
↳ J. J. Wich

* Topic 7 :- FUNCTION AND RECURSION

(1) #write a function power (a, b) to calculate the value of a raised to b in c.

```
#include <stdio.h>
int power (int a, int b) {
    int result = 1;
    for (int i = 0; i < b; i++) {
        result *= a; }
    return result; }
int main() {
    int base, exponent;
    scanf("%d %d", &base, &exponent);
    int result = power (base, exponent);
    printf("%d raise %d is %d", base
    exponent, result);
    return 0; }
```

(2) function to determine leap year or not.

```
#include <stdio.h>
int isleapyear (int year) {
if ((year % 4 == 0 && year % 100
    != 0) || (year % 400 == 0)) {
    return 1;
} else { return 0; } }
int main () {
```

```
int year;
printf("Enter a year: ");
scanf("%d", &year);
if (isleapyear(year)) {
printf("%d is a leap year", year);
else {printf("%d is not a leap
year \n", year); }
return 0;
}
```

(3) Write a recursive function to calculate factorial of a number in C.

```
#include <stdio.h>
int factorial(int n) {
if (n == 0 || n == 1)
return 1;
else
return n * factorial(n-1); }
int main() {
    int num;
    printf("Enter a number:");
    scanf("%d", &num);
    if (num<0)
    printf("not defined");
    else
    printf("Factorial of %d is %d\n",
            num, factorial(num));
    return 0;
}
```

(4) Write a function to swap two integers using call by value. Show that the original values are not changed in C.

```
#include <stdio.h>
Void swap (int a, int b) {
    int temp = a;
        a = b;
        b = temp;
printf ("a = %d, b = %d", a, b);
}
int main () {
    int x = 10, y = 20;
  printf("Before swap : x = %d, y = %d",
        x, y);
  swap (x, y);
  printf ("x = %d, y = %d", x, y);
  return 0; }
// value remain unchanged
```

(5) Simple calculator :-

```
#include <stdio.h>
float add (float a, float b) {
    return a+b;
}
  float subtract (float a, float b) {
    return a-b;
}
Float multiply (float a, float b) {
    return a*b; }
```

```
float divide (float a, float b) {
    if (b != 0)
        return a / b;
    else { printf("error"); return 0; } }
int main() {
    float num1, num2, result;
    char z;
    printf("enter expression");
    scanf("%f %c %f", &num1, &z, &num2);
    switch (z) {
        case '+': result = add (num1, num2);
        break;
        case '-': result = subtract (num1, num2); break;
        case '*':
            result = multiply (num1, num2);
            break;
        default:
            printf ("Invalid operator\n");
            return 1;
    }
printf ("Result: %0.2f %c %0.2f = %0.2f \n", num1, z, num2, result);
    return 0;
    }
```

* Topic 8 :- Structures and unions

(1) →

```
#include <stdio.h>
#include <string.h>
struct student {
    int rollnumber;
    char name[50];
    char coursename[50];
    char majorsubject[50];
    char minorsubject[50];
    };
void names(struct student students[],
int size) {
  printf("List of student Names:\n");
  for(int i=0; i<size; i++) {
   printf("%d. %s\n", i+1, students[i].
   name); } }
void printstudentR(struct student
  students[], int size, int roll) {
  int found = 0;
  for(int i=0; i<size; i++) {
   if (students[i].rollnumber == roll) {
   printf("Roll Number: %d\n", students[i]
.rollnumber);
   printf("Name : %s\n", students[i].
name);
  printf("course name: %s", students[i].
   coursename);
   printf("major subject: %s\n", students
      [i].majorsubject);
  printf("minorsubject: %s\n", students
      [i].minorsubject);
```

```
        found = 1; break; } }
    if (!found) {
    printf("No student found with roll
number %d\n", roll); } }
    int main () {
    struct student students [10];
    for (int i=0; i<10; i++) {
    printf("Enter details for student %d
    , i+1);
    printf ("Roll number: ");
    scanf ("%d", &students[i].rollnumber
    getchar();
     fgets (s[i].name, 30, stdin);
      return 0; }
```

(2) →

```
#include <stdio.h>
struct customer {
    int acc;
    char name[30];
    float bal; };
void lowBalance(struct customer c[],
int n) {
    for(int i=0; i<n; i++)
    if (c[i].bal<100)
    printf("Acc:%d, Name:%s\n",
c[i].acc, c[i].name); }
void updatebalance(struct customer c[],
int n, int acc, float amt, int code) {
    for(int i=0; i<n; i++) {
        if (c[i].acc==acc) {
            if (code==1)
            c[i].bal+=amt;
        else if (amt>c[i].bal)
    printf("insufficient");
    else c[i].bal-=amt; } } }
    int main() {
        struct customer c[10];
        for(int i=0; i<10; i++) {
        printf("customer %d:\n", i+1);
        scanf("%d %s %f", &c[i].acc,
    c[i].name, &c[i].bal); }
    printf("customers with balance<Rs.100"
); low balance(c, 10);
    int acc, code;
    float amt;
printf("Enter Account NO, Amount, code
    (1=deposit, 0=withdraw):");
```

```
scanf ("%d %f %d", &acc, &amt, &
);
updateBalance (c, ID, acc, amt, code
return 0;
}
```

(3) →

```
#include <stdio.h>
struct students_data{
    int roll;
    char name [30];
    float phy, math, chem, total;
};
int main (){
    struct student_data S;
printf ("Enter Roll No, Name, marks of
phy, maths, chemistry: \n");
scanf ("%d %s %f %f %f", &S.roll
S.name, &S.phy, &S.math, &S.chem);
S.total = S.phy + S.math + S.chem;
printf ("given below %d, %s, %f, %f,
%f ", S.roll, S.name, S.phy, S.math,
S.chem, S.total);
return 0;
}
```

(4) →

```
#include <stdio.h>
#include <string.h>
struct cricketer {
    char name[50];
    int age;
    int matches;
    float avg_runs; };
int main() {
    struct cricketer c[10] = {
{ "sachin", 47, 200, 59.78 },
{ "Dravid", 45, 164, 52.37 },
{ "Kohli", 34, 111, 49.29 },
{ "Dhoni", 42, 90, 38.09 },
{ "sehwag", 44, 104, 49.34 },
{ "Laxman", 48, 134, 45.97 },
{ "Ganguly", 50, 113, 42.17 },
{ "Rohit", 36, 52, 46.54 },
{ "Pant", 25, 33, 43.67 },
{ "Ashwin", 37, 94, 27.18 } };
// Sorting
for(int i=0; i<9; i++) {
  for(int j=i+1; j<10; j++) {
if ( c[i].avg_runs > c[j].avg_runs){
  struct cricketer temp = c[i];
    c[i] = c[j];
    c[j] = temp; } }
printf("Sorted avg. runs:\n");
for(int i=0; i<10; i++) {
  printf("%-10s\t Age: %d\t AVG Runs:
%.2f\n", c[i].name, c[i].age,
c[i].matches, c[i].avg_runs; }
```

* Topic 9 :- File Handling

(1) Read a file and count no. of lines, words, characters, spaces in it.

```
#include <stdio.h>
#include <ctype.h>
int main () {
 FILE * file;
 char ch;
 int lines = 0, words = 0, characters = 0,
spaces = 0;
 int inword = 0;
 file = fopen ("example.txt", "r");
 if (file == NULL) {
 printf (" Error opening file. \n");
 return 1; }
while ( cch = fgetc(file) ! = NULL) {
 characters ++;
 if (ch == '\n') {
    lines ++; }
 if (isspace (ch)) {
    if  (ch == ' ')
      spaces ++; in word = 0; }
 else { if (! inword) {
      words ++;
      inword = 1; } } }
 fclose (file);
printf ("%d, %d %d %d", lines, a
 words, characters, spaces);
  }
```

(2) Instead of characters, accept one line at a time from the user and write it into the file.

```
#include <stdio.h>
int main() {
    FILE *file;
    char line[1000];
    file = fopen("output.txt", "w");
    if (file == NULL) {
    printf("NO CONTENT DETECTED");
    return 1; }
    printf("Enter lines of text");
    while (1) {
      gets(line);
    if (strcmp(line, "exit") == 0) {
    break; }
    fputs(line, file);
    fputs("\n", file); }
    fclose(file);
    printf("Successfully written");
    return 0; }
```

(3) Read a content of the file line by line.

```
#include <stdio.h>
int main() {
    FILE *file;
    char line[1000];
    file = fopen("example.txt", "r");
```

```
if (file == NULL) {
printf ("NO CONTENT DETECTED");
return 1; }
printf ("contents :-");
while (fgets (line, sizeof (line), fi
) != NULL) {
printf ("%s", line);
fclose (file);
return 0; }
```

(4) Record of Roll number and name storing to excel.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main () {
    FILE *file;
    int rollNO;
    char name[100];
    char choice;
file = fopen ("roll.csv", "w");
if (file == NULL) {
printf("NO CONTENT");
return 1; }
fprint (file, "Roll NO, Name \n");
do { printf ("Enter Roll NO : ");
scanf ("%d", &rollNO);
getchar ();
printf ("Enter name : ");
fgets (name, sizeof (name), stdin);
```

```
fprint (file, "%d,%s\n", rollno,
  name);
printf("Do you want to enter
another record ");
  scanf("%c", &choice);
  getchar(); }
  while (choice == 'y' || choice == 'Y');
  fclose (file);
  system("start excel.exe roll.csv");
  return 0;
  }
```