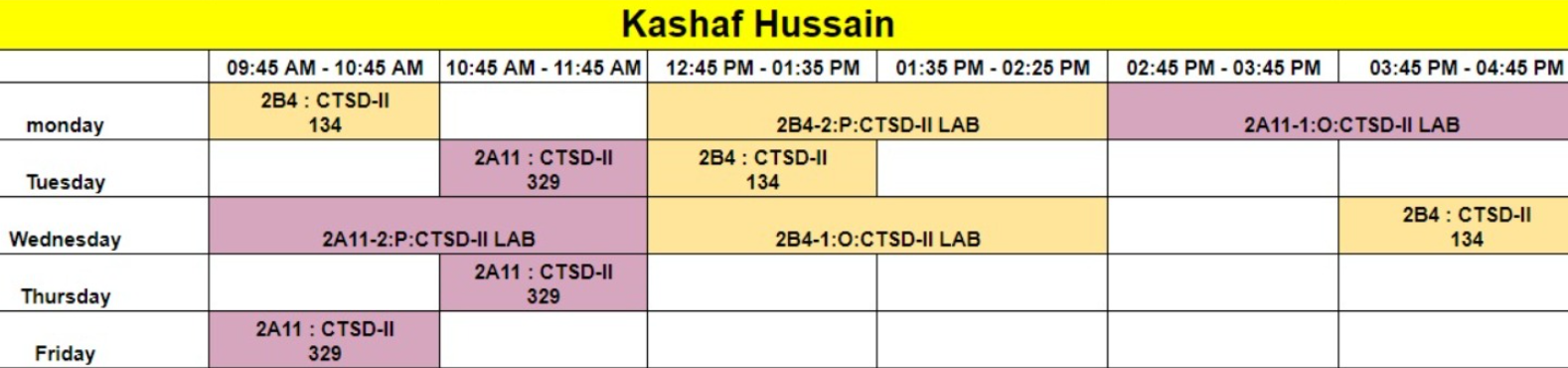
Parul University

Computational Thinking & Structured Design

LAB work

**C & Data Structures**



**Week 1:**

a) Write a C program to increase or decrease the existing size of an 1D array?

b) Write a C program on 2D array to Increase & Decrease

i) No of subarrays

ii) elements in the subarrays

#include<stdio.h>

#include<stdlib.h>

main()

{

int \*p,n,i,newlength;

printf("enter the size of array : ");

scanf("%d",&n):

p=(int\*)malloc(n\*sizeof(int));

if(p==NULL)

{

printf("No memory allocated : “);

exit(1);

}

printf("enter the data : ");

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

scanf("%d",&p[i]);

printf("values are\n");

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

printf("enter new size : ");

scanf("%d",&newlenght);

p=relloc(p,n\*sizeof(int));

printf("enter values : ");

for(i=0;i<newlength;i++)

scanf("%d",&p[i]);

printf("values are : ");

for(i=0;i<newlength;i++)

printf("%d",p[i]);

free(p);

}

/\* write a program to increase or decrease the size of 1D array. \*/

#include<stdio.h>

#include<stdlib.h>

int main()

{

int n,i,newsize,\*p;

printf("Enter the size of Dynamic array : ");

scanf("%d",&n);

p = (int \*) malloc(n\*sizeof(int));

if(p==NULL)

{

printf("Memory NOT allocated : ");

exit(1);

}

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

{

printf("Enter the data : ");

scanf("%d",(p+i));

}

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

{

printf("%3d",\*(p+i));

}

printf("Enter the new size of Dynamic array : ");

scanf("%d",&newsize);

p = (int \*) malloc(newsize\*sizeof(int));

for(i=0;i<newsize;i++)

{

printf("Enter the data : ");

scanf("%d",(p+i));

}

for(i=0;i<newsize;i++)

{

printf("%3d",\*(p+i));

}

return 0;

}

#include<stdio.h>

#include<stdlib.h>

int print(int n, int m, int\* ptr){

for(int i = 0; i < n; i++){

for(int j = 0; j < m; j++){

printf("Enter the elements = ");

scanf("%d",&(ptr[i \* m + j]));

}

printf("\n");

}

printf("The 2D Array = \n");

for(int i = 0; i < n; i++){

for(int j = 0; j < m; j++){

printf("%d ",ptr[i \* m + j]);

}

printf("\n");

}

}

int main(){

int n,m;

printf("Enter the number of rows = ");

scanf("%d",&n);

printf("Enter the number of columns = ");

scanf("%d",&m);

int \*ptr = (int \*) malloc((n\*m)\*sizeof(int));

print(n,m,ptr);

ptr = (int \*) realloc(ptr,(n\*m)\*sizeof(int));

print(n,m,ptr);

free(ptr);

return 0;

}

**Week-02**

#include <stdio.h>

#include <time.h>

void main(void)

{

// Get the current time

time\_t current\_time = time(NULL);

// Convert the time to a string using the desired format

char date\_string[20];

strftime(date\_string, 20, "%Y-%m-%d", localtime(&current\_time));

// Print the date string

printf("The current date is: %s\n", date\_string);

}

#include <stdio.h>

#include <time.h>

void main(void)

{

// Get the current time

time\_t current\_time = time(NULL);

time\_t system\_time = time(NULL);

// Convert the time to a string using the desired format

char date\_string[20];

strftime(date\_string, 20, "%Y-%m-%d", localtime(&current\_time));

printf("The current date is: %s\n", date\_string);

strftime(date\_string, 20, "%d-%m-%Y", localtime(&current\_time));

// Print the date string

printf("The current date is: %s\n", date\_string);

time(&system\_time);

printf("The current time is: %s\n", ctime(&system\_time));

}

#include<stdio.h>

// This is macro definition

#define PI 3.14

void main()

{

// declaration and initialization of radius

int radius = 5;

// declaration and calculating the area

float area = PI \* (radius\*radius);

// Printing the area of circle

printf("Area of circle is %f", area);

}

/\* C program to demonstrate the use of conditional directives. \*/

#include <stdio.h>

#define NUM 15

#if NUM > 200

#undef NUM

#define NUM 200

#elif NUM < 50

#undef NUM

#define NUM 50

#else

#undef NUM

#define NUM 100

#endif

void printValue(int value)

{

printf("%d", value); }

int main()

{

printValue(NUM); // NUM = 50

return 0;

}

// C program to illustrate the use of Pragma to print the custom message

#include <stdio.h>

// not defining KashifHussain to trigger pragma message

// #define KashifHussain

int main()

{

#ifndef KashifHussain

#pragma message(" KashifHussain is not defined.")

#endif

printf("Hello Kashif !\n");

return 0;

}

**Week-03**

a) Write a C program that uses functions to perform the following Operations

i) Reading a complex number

ii) Writing a complex number

iii) Addition of two complex numbers

iv) Multiplication of two complex numbers

// addition of complex numbers

#include <stdio.h>

// define a structure for complex number

typedef struct complexNumber {

int real;

int img;

} complex;

// This function accepts two complex numbers

// as parameter and return addition of

// them.

complex add(complex x, complex y);

// Driver code

int main()

{

// Define three complex type numbers

complex a, b, sum;

// First complex number

a.real = 2;

a.img = 3;

// Second complex number

b.real = 4;

b.img = 5;

// Print first complex number

printf("\n a = %d + %di", a.real, a.img);

// print second complex number

printf("\n b = %d + %di", b.real, b.img);

// call add(a,b) function and

// pass complex numbers a & b

// as an parameter.

sum = add(a, b);

// Print result

printf("\n sum = %d + %di", sum.real, sum.img);

return 0;

}

// Complex add(complex x, complex y)

// function definition

complex add(complex x, complex y)

{

// Define a new complex number.

complex add;

// Add real part of a&b

add.real = x.real + y.real;

// Add Imaginary part of a&b

add.img = x.img + y.img;

return (add);

}

b) Write a c program to store records of n students

based on roll\_no, name, gender and 5 subject marks

i) calculate percentage each student using 5 subjects

ii) Display the student list according to their percentages.

1. Write a program to read the records of ‘n’ students and sort them on roll number using structure.

#include<stdio.h>

#include<stdlib.h>

struct student

{

char name[10];

int age;

int roll;

};

void main()

{

struct student arr[20], temp;

int i,j,n;

n = 5;

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

{

printf("\nEnter the name, age, roll : ");

scanf("%s %d %d",&arr[i].name,&arr[i].age,&arr[i].roll);

}

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

{

printf("%s\t %d\t %d \n",arr[i].name,arr[i].age,arr[i].roll);

}

/\* sorting \*/

for(i = 1; i<n ;i++)

{

for(j = 0;j<n-i; j++)

{

if(arr[j].roll > arr[j+1].roll)

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1]= temp;

}

}

printf("\n");

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

printf("\n%s \t %d \t %d", arr[i].name, arr[i].age, arr[i].roll);

}