[Skip to main content](javascript:void(0);)

[VIRENDRA KUMAR SINGH](https://cuchd.blackboard.com/ultra/profile)

* [Activity Stream](https://cuchd.blackboard.com/ultra/stream)
* [Courses](https://cuchd.blackboard.com/ultra/course)
* [Organizations](https://cuchd.blackboard.com/ultra/organization)
* [Calendar](https://cuchd.blackboard.com/ultra/calendar)
* [Messages](https://cuchd.blackboard.com/ultra/messages)
* [Grades](https://cuchd.blackboard.com/ultra/grades)
* [Tools](https://cuchd.blackboard.com/ultra/tools)
* [Sign Out](https://cuchd.blackboard.com/ultra/logout)

[Privacy](https://help.blackboard.com/1000en_US)[Terms](https://help.blackboard.com/1001en_US)

Courses

Current Courses

Upcoming Courses

View Course ListView Course Grid



**Filter**

All Courses

25

items per page

**Favorites**

**[AU2\_Events](javascript:void(0);)**

**AU2\_Events**

Harjeet Kaur

**[Biology for Engineers](javascript:void(0);)**

**20SZT-148\_20BCS39\_All**

Amanpreet Kaur

**[COMMUNICATION SKILLS](javascript:void(0);)**

**20UCT-141\_20BCS39\_ALL**

Harpreet Kaur

**[COMPUTATIONAL MATHEMATICS](javascript:void(0);)**

**20SMT-125\_20BCS39\_B**

Manpreet Kaur

**[Communication Skills Lab](javascript:void(0);)**

**20UCP-142\_20BCS39\_B**

Harpreet Kaur

**[Computer Graphics using CAD Lab](javascript:void(0);)**

**20MEP-114\_20BCS39\_B**

Dipesh Popli

**[DIGITAL ELECTRONICS LAB](javascript:void(0);)**

**20ECP-116\_20BCS39\_B**

Kuldip Kumar

**[Digital Electronics](javascript:void(0);)**

**20ECT-115\_20BCS39\_All**

Kuldip Kumar

**[FOUNDATIONS OF AI APPLICATIONS LAB](javascript:void(0);)**

**20ECP-118\_20BCS39\_B**

Ankita Aggarwal

**[LIFE SKILLS AND MENTORING-1](javascript:void(0);)**

**20UCY-146\_20BCS39\_B**

Harpreet Kaur

**[PROBLEM SOLVING WITH PROGRAMMING](javascript:void(0);)**

**20CST-111\_20BCS39\_B**

Snigdha Luthra

**[PROBLEM SOLVING WITH PROGRAMMING LAB](javascript:void(0);)**

**20CSP-112\_20BCS39\_B**

Snigdha Luthra

×

Top of Form

20CSP-112\_20BCS39\_B

PROBLEM SOLVING WITH PROGRAMMING LAB

PROBLEM SOLVING WITH PROGRAMMING LAB

Bottom of Form

[Skip to main content](javascript:void(0);)

**Snigdha Luthra**

Instructor

**Details & Actions**

**[Roster](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/roster)**

[View everyone in your course](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/roster)

**Blackboard Collaborate**

Join session

**[Attendance](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/attendanceGrade?courseId=_10513_1)**

[View your attendance](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/attendanceGrade?courseId=_10513_1)

**[Groups NEW](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/course-group-enroll?courseId=_10513_1)**

[View groups to join](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/course-group-enroll?courseId=_10513_1)

**[Announcements](javascript:void(0);)**

[View archive](javascript:void(0);)

**[Books & Tools](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/booksAndTools?parentId=_1405906_1)**

[View course & institution tools](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/booksAndTools?parentId=_1405906_1)

**[OneDrive](javascript:void(0);)**

[Connect account](javascript:void(0);)

**Course Content**

[**Meet Your Instructor Virtually**](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/edit/document/_1424322_1?courseId=_10513_1)

**About this Course**

This section contains the overview and objectives of the lab course. This course will make the students of all engineering streams to have good hands on knowledge of programming concepts in C language.

**Unit 1: Introduction to basic building blocks to programming**

This section consists of basic programming problems based on various building blocks of C Programming.

**Unit 2: Dealing with real-world problems**

This section gives hands on experience for solving various programming problems using looping constructs, arrays, strings and functions.

**Unit 3: Handling heterogeneous data and memory Management**

This section gives hands on experience for solving various programming problems using Pointers, Structures and Dynamic memory allocation.

**Practical 3.1**

This practical is based on the usage of Pointers in C.

[**Practical 3.1 Manual**](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/edit/document/_3624528_1?courseId=_10513_1)

This document contains the code & execution result of practical 3.1.

[**Pre-lab Questions**](javascript:void(0);)

Due date: 10/9/20, 12:00 AM

[**Practical 3.1 Video/Study Material**](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/edit/document/_3624530_1?courseId=_10513_1)

[**PRACTICAL 3.1.1 & 3.1.2 PPT**](javascript:void(0);)

[**PRACTICAL 3.1.3 & 3.1.4 PPT**](javascript:void(0);)

[**Worksheet**](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/edit/document/_3624533_1?courseId=_10513_1)

WORKSHEET TEMPLATE

[**Post-lab Questions**](javascript:void(0);)

Due date: 10/9/20, 12:00 AM

[**Discussion Forum Practical 3.1**](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/discussion/_3624523_1?courseId=_10513_1)

Have you heard of “Partial Initialization of Array” term? What does it mean? Justify your answer with an example.

[**Journal Practical 3.1**](https://cuchd.blackboard.com/ultra/courses/_10513_1/outline/journal/_3624536_1?courseId=_10513_1)

Due date: 1/10/21, 6:31 PM

No more content items to load

**Practical 3.2**

This practical covers the use of Structures in C programming language.

**Practical 3.3**

This practical is based on DMA. It incorporates how different DMA functions can be used.

No more content items to load

No more content items to load

×

PROBLEM SOLVING WITH PROGRAMMING LAB

Practical 3.1 Manual

Practical 3.1 Manual

* **SUB PROGRAMS WITH SOLUTIONS-**

**Practical 8.1:** WAP to read an array of elements and print the same in the reverse order along with their addresses using pointer.

**Solution:**

#include <stdio.h>

int main()

{

  int n, i, arr1[15];

  int \*pt;

  printf("\n\n Pointer : Print the elements of an array in reverse order :\n");

  printf("----------------------------------------------------------------\n");

  printf(" Input the number of elements to store in the array (max 15) :\n ");

  scanf("%d",&n);

  pt = &arr1[0]; // pt stores the address of base array arr1

  printf(" Input %d number of elements in the array : \n",n);

  for(i=0;i<n;i++)

   {

            printf(" element - %d : ",i+1);

            scanf("%d",pt);//accept the address of the value

            pt++;

           }

  pt = &arr1[n - 1];

  printf("\n The elements of array in reverse order are :");

  for (i = n; i > 0; i--)

  {

     printf("\n element - %d : %d ", i, \*pt);

     pt--;

  }

  printf("\n\n");

  return 0;

}

**OUTPUT**

**Practical 8.2:** Write a function code that is returning pointer to the larger value out of two passed values.

**Solution:**

#include <stdio.h>

int \*getMax(int \*, int \*); //function declaration

int main(void) {

 // integer variables

 int x, y;

 printf(" FIND MAX VALUE\n Enter two integers-\n ");

 scanf("%d%d", &x,&y);

 // pointer variable

 int \*max = NULL;

 /\*\*

  \* get the variable address that holds the greater value

  \* for this we are passing the address of x and y

  \* to the function getMax()

  \*/

 max = getMax(&x, &y);

 // print the greater value

 printf(" Max value: %d\n", \*max);

 return 0;

}

// function definition

int \*getMax(int \*m, int \*n) {

 /\*\*

  \* if the value pointed by pointer m is greater than n

  \* then, return the address stored in the pointer variable m

  \*/

 if (\*m > \*n) {

   return m;

 }

 /\*\*

  \* else return the address stored in the pointer variable n

  \*/

 else {

   return n;

 }

}

**OUTPUT**

***Practical 8.3:*** The bank balance of N persons of a city are recorded. Due to COVID-19 government has decided to credit accounts with rupees 1000 of all those persons whose balance is NULL , Write a function Update\_balace() by passing pointer to an array as argument and print the updated bank balance list in calling function.

**Solution:**

#include<stdio.h>

#include<conio.h>

void Update\_balance(int \*p, int a[], int i)

{

   p= a[i];

           printf(" Amount credited is %d\n", p);

}

int main()

{

   int a[100];

   int n,i,\*p;

   printf(" Enter number of persons\n ");

   scanf("%d", &n);

           p= &a[0];

           for(i=0;i<n;i++)

   {

       printf(" Enter current amount of person %d in account- ",i+1);

       scanf("%d",p); // bank balance of n persons is stored in array

       p++;

   }

   p = &a[0];

   for (i=0;i<n;i++)

   {

       if(\*p==0)  // checking for those persons among n whose balance is either 0 or lesser

       {

           \*p = \*p + 1000;

                          printf("\n Amount is credited to person %d \n", i+1);

           Update\_balance(&p,a,i);

        }

        p++;

           }

           return 0;

}

**OUTPUT**

**Practical 8.4:** The CGPA of 5 semesters of N students has stored in NX5 array and names of corresponding students are stored in separate string array. The student who got average CGPA >=8 is eligible for placements .Pass pointers to both array to function not\_eligible() and print list of non-eligible students.

**Solution:**

#include<stdio.h>

struct sp

{

           char name[100];

}s[100];

int eligible(double b,struct sp s[],int k)

{

           printf("\nstudent : %s and c.g.p.a : %0.2lf elgible\n",s[k].name,b);

}

int not\_eligible(double b,struct sp s[],int k)

{

           printf("\nstudent : %s and c.g.p.a : %0.2lf not elgible\n",s[k].name,b);

}

int main()

{

           int n,i,j;

           double a[100][5];

           printf(" Enter number of students- ");

           scanf("%d",&n);

           double sum[n],cgpa[n];

           for(i=0;i<n;i++)

           {

                       printf(" %d. Enter student name- ",i+1);

                       scanf("%s",&s[i].name);

                       for(j=0;j<5;j++)

                       {

                                   printf(" Enter CGPA (0-10) in sem %d- ",j+1);

                                   scanf("%lf", &a[i][j]);

                       }

           }

           for(i=0;i<n;i++)

           {

                       sum[i]=0;

                       for(j=0;j<5;j++)

                       {

                                   sum[i]= sum[i]+ a[i][j];

                       }

                       cgpa[i]= (double)sum[i]/5.0;

                       if(cgpa[i] >=8)

                       {

                                   eligible(cgpa[i],s,i);

                       }

                       else

                       {

                                   not\_eligible(cgpa[i],s,i);

                       }

           }

           return 0;

}

**OUTPUT**

[Help for current page](https://ultra.content.blackboardcdn.com/ultra/uiv3900.2.0-rel.4_6d79cf3" \t "_blank)

Courses loaded successfully