LALIT NARAYAN MITHILA UNIVERSITY

(KAMESHWARANAGAR, DARBHANGA)

APPLICATION FORM FOR ADMISSION IN UG SESSION (2021-2024)

Unique Form ID :	**212542948**	**Course Applied (HONOURS) : Bachelors of Arts (Political Science)**	Aman Kumar Rajak **Signature of the Applicant**

PERSONAL DETAILS

Name :	**AMAN KUMAR RAJAK**	**Course Name :**	Bachelors of Arts (Political Science)
Application No :	**2101151702**	**Course Type :**	HONOURS
Father's Name :	AJAY KUMAR SAFI	**Gender :**	MALE
Mother's Name :	RENU KUMARI	**Category :**	SC
Date of Birth	16/01/2003	**Religion :**	HINDU
Marital Status :	Unmarried	**Blood Group :**	O+
Aadhar :	712295324112	**Identification Marks:**	TWO MOLES ON LEFT CHEEK
Email :	amanraz0709@gmail.com	**Mobile Number :**	6299542755
Permanent Address :	VILLAGE- PARSA BISHANPUR POST OFFIC- PARSA BISHANPUR POLICE STATION- KEOTI		
Present Address :	VILLAGE- PARSA BISHANPUR POST OFFIC- PARSA BISHANPUR POLICE STATION- KEOTI		
State :	BIHAR	**District :**	DARBHANGA
Pin Code :	847337	**Parent Mobile Number :**	8789565621

PAYMENT DETAILS

Reference Number :	47738716	**Paid Amount :**	400.00
Payment Mode :	ONLINE	**Transaction Date :**	09/08/2021 08:37:54
Order ID :	2101151702_419834	**Registration Date :**	08/08/2021 10:50:10

QUALIFICATION DETAILS

Examination	Status	Board/University	Passing Year	Marks Obt.	Total Marks	Percentage	Stream
Intermediate or Equivalent	Passed	CBSE	2021	403	500	80.6	Science

COLLEGE PREFERENCE AND SUBJECT DETAILS

	College Name	Subsidiary I	Subsidiary II	Language
College Preference 1	202 -C. M. College, Darbhanga	History	English	RB Non Hindi with English
College Preference 2	210 -M. L. S. M. College, Darbhanga	History	English	RB Non Hindi with English
College Preference 3	223 -M. G. College, Darbhanga	History	English	RB Non Hindi with English
College Preference 4	232 -R. B. J. College, Bela, Darbhanga	History	English	RB Non Hindi with English
College Preference 5	212 -Marwari College, Darbhanga	History	English	RB Non Hindi with English

I hereby declare that the particulars filled are true, correct and complete to the best of my knowledge and belief and nothing has been concealed.

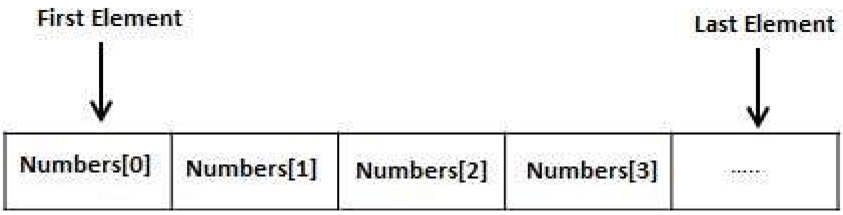
C - ARRAYS

http://www.tutorialspoint.com/cprogramming/c_arrays.htm

C programming language provides a data structure called **the array**, which can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. A specific element in an array is accessed by an index.

All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.



Declaring Arrays

To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows:

```
type arrayName [ arraySize ];
```

This is called a *single-dimensional* array. The **arraySize** must be an integer constant greater than zero and **type** can be any valid C data type. For example, to declare a 10-element array called **balance** of type double, use this statement:

```
double balance[10];
```

Now *balance* is avariable array which is sufficient to hold upto 10 double numbers.

Initializing Arrays

You can initialize array in C either one by one or using a single statement as follows:

```
double balance[5] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

The number of values between braces { } can not be larger than the number of elements that we declare for the array between square brackets [ ].

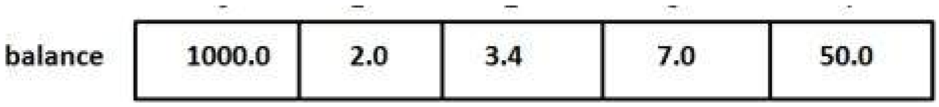
If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write:

```
double balance[] = {1000.0, 2.0, 3.4, 7.0, 50.0};
```

You will create exactly the same array as you did in the previous example. Following is an example to assign a single element of the array:

```
balance[4] = 50.0;
```

The above statement assigns element number 5th in the array with a value of 50.0. All arrays have 0 as the index of their first element which is also called base index and last index of an array will be total size of the array minus 1. Following is the pictorial representation of the same array we discussed above:



balance	1000.0	2.0	3.4	7.0	50.0

Accessing Array Elements

An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example:

```
double salary = balance[9];
```

The above statement will take 10th element from the array and assign the value to salary variable. Following is an example which will use all the above mentioned three concepts viz. declaration, assignment and accessing arrays:

```
#include <stdio.h>

int main ()
{
   int n[ 10 ]; /* n is an array of 10 integers */
   int i,j;

   /* initialize elements of array n to 0 */
   for ( i = 0; i < 10; i++ )
   {
      n[ i ] = i + 100; /* set element at location i to i + 100 */
   }

   /* output each array element's value */
   for (j = 0; j < 10; j++ )
   {
      printf("Element[%d] = %d\n", j, n[j] );
   }

   return 0;
}
```

When the above code is compiled and executed, it produces the following result:

```
Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109
```

C Arrays in Detail

Arrays are important to C and should need lots of more details. There are following few important concepts related to array which should be clear to a C programmer:

Concept	Description
Multi-dimensional arrays	C supports multidimensional arrays. The simplest form of the multidimensional array is the two-dimensional array.
Passing arrays to functions	You can pass to the function a pointer to an array by specifying the array's name without an index.

Return array from a function	C allows a function to return an array.
Pointer to an array	You can generate a pointer to the first element of an array by simply specifying the array name, without any index.