

PROGRAMMING FOR DATA STRUCTURES AND ALGORITHMS

II semester of B.Tech

As per the curricullam and syllabus
Of

Bharath Institute of Higher Education & Research

PREPARED BY

DR. YOGESH RAJKUMAR

NEW EDITION

Department of Information Technology



Bharath
INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Declared as Deemed - to - be - University under section 3 of UGC Act 1956)
ACCREDITED WITH 'A' GRADE BY NAAC

PROGRAMMING FOR DATA STRUCTURES AND ALGORITHMS

II semester of B.Tech

As per the curricullam and syllabus
Of

Bharath Institute of Higher Education & Research

PREPARED BY

DR. YOGESH RAJKUMAR

Department of Information Technology



Bharath

INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Declared as Deemed - to - be - University under section 3 of UGC Act 1956)

ACCREDITED WITH 'A' GRADE BY NAAC

SCHOOL OF COMPUTING

DEPARTMENT OF INFORMATION TECHNOLOGY

LAB MANUAL

SUBJECT NAME: PROGRAMMING FOR DATA STRUCTURES AND

ALGORITHMS

SUBJECT CODE: U20ITCJ01

Regulation - 2020

VISION AND MISSION OF THE INSTITUTE

VISION

“Bharath Institute of Higher Education & Research (BIHER) envisions and constantly strives to provide an excellent academic and research ambience for students and members of the faculties to inherit professional competence along with human dignity and transformation of community to keep pace with the global challenges so as to achieve holistic development.”

MISSION

- To develop as a Premier University for Teaching, Learning, Research and Innovation on par with leading global universities.
- To impart education and training to students for creating a better society with ethics and morals.
- To foster an interdisciplinary approach in education, research and innovation by supporting lifelong professional development, enriching knowledge banks through scientific research, promoting best practices and innovation, industry driven and institute-oriented cooperation, globalization and international initiatives.
- To develop as a multi-dimensional institution contributing immensely to the cause of societal advancement through spread of literacy, an ambience that provides the best of international exposures, provide health care, enrich rural development and most importantly impart value-based education.
- To establish benchmark standards in professional practice in the fields of innovative and emerging areas in engineering, management, medicine, dentistry, nursing, physiotherapy and allied sciences.
- To imbibe human dignity and values through personality development and social service activities.

VISION AND MISSION OF THE DEPARTMENT

VISION

To be an excellence in education and research in Information Technology producing global scholars for improvement of the society

MISSION

- To provide sound fundamentals, and advances in Information Technology, Software Engineering, data Communications and Computer Applications by offering world class curriculum.
- To create ethically strong leaders and expert for next generation IT.
- To nurture the desire among faculty and students from across the globe to perform outstanding and impactful research for the benefit of humanity and, to achieve meritorious and significant growth.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

The Program Educational Objectives (PEOs) of Information technology are listed below: The graduate after 3-5 years of programme completion will

PEO1: PREPARATION

To provide students with sound fundamental in Mathematical, Scientific and Engineering fundamentals necessary to formulate, analyse, and comprehend the fundamental concepts essential to articulate, solve and assess engineering problems and to prepare them for research & development and higher learning.

PEO2: CORE COMPETENCE

To apply critical reasoning, quantitative, qualitative, designing and programming skills, to identify, solve problems and to analyze the experimental evaluations, and finally making appropriate decisions along with knowledge of computing principles and applications and be able to integrate this knowledge in a variety of industry and inter-disciplinary setting.

PEO3: PROFESSIONALISM

To broaden knowledge to establish themselves as creative practicing professionals, locally and globally, in fields such as design, development, problem solving to production support in software industries and R&D sectors.

PEO4: SKILL

To provide better opportunity to become a future researchers / scientist with good communication skills so that they may be both good team-members and leaders with innovative ideas for a sustainable development.

PEO5: ETHICS

To be ethically and socially responsible solution providers and entrepreneurs in Computer Science and other engineering discipline.

PROGRAMME OUTCOMES

PO 1	Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem Analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO 3	Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions for complex problems.
PO 5	Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO 6	The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO 7	Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9	Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	Lifelong Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOME

PSO 1	Programming Design : Design and develop algorithm for real life problems using latest technologies and solve it by using computer programming languages and database technologies .
PSO 2	IT Business Scalable Design : Analyze and recommend computing infrastructures and operations requirements and Simulate and implement information networks using configurations, algorithms, suitable protocol and security for valid and optimal connectivity.
PSO 3	Intelligent Agents Design : Design and execute projects for the development of data modeling, data analytics and knowledge representation in various domain.

U20ITCJ01 – PROGRAMMING FOR DATA STRUCTURES AND ALGORITHMS

PART-A INTRODUCTION OF THE COURSE

Data Structure is the group of data elements which provides an efficient way of storing and organizing data in the computer so that it can be used efficiently

U20ITCJ01	PROGRAMMING FOR DATA STRUCTURES AND ALGORITHMS	L	T	P	C
	Contact Hours–60	2	0	2	3
	Pre-requisite – U20ITCJ01				
	CourseDesignedby–Department of Information Technology				
OBJECTIVES:ToenhancethepracticalknowledgeonwritingprogramsusingPython					
NAME OF THE COORDINATOR	Dr.Yogesh Rajkumar				

COURSE OUTCOMES(COs)															
CO1	Demonstrate the basics of data structures and various algorithms.														
CO2	Experiment with Various Linear ADTs - Lists, stacks, queues and its applications.														
CO3	Construct the trees with its representations and methods.														
CO4	Compare various sorting algorithms, properties, and its methods														
CO5	Interpret hashing concepts and its techniques														
CO6	Justify the non-linear data structures graph with its various methods.														
MAPPING BETWEEN COURSE OUTCOMES & PROGRAM OUTCOMES(3/2/1 INDICATES STRENGTH OF CORRELATION-3-High, 2-Medium,1-Low)															
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	3	-	3	1				1	1	-	3	3	2	2
CO2	2	3	1	3	1				2	1	-	3	3	2	2
CO3	2	3	2	3	1				2	1	-	3	3	2	2
CO4	2	3	2	3	1				2	1	-	3	3	2	2
CO5	1	3	--	3	1				1	1	-	3	3	2	2
CO6	3	3	2	3	1				2	1	-	3	3	2	2
Category	Engineering Sciences(ES)														
Approval	47 th Academic Council Meeting held in Aug, 2020														

CONTENT

S.NO	NAME OF THE EXPERIMENT	PAGE NO
1	Compute the GCD of two numbers.	6
2	Find the square root of a number (Newton's method)	8
3	Exponentiation (Power of a number)	10
4	Find the maximum of a list of numbers	12
5	Linear search and Binary search	14
6	Selection sort	16
7	Insertion sort	19
8	First n prime numbers	21
9	Multiply matrices	23
10	Find the most frequent words in a text read from a file	25

Ex. No1

COMPUTE THE GCD OF TWO NUMBERS.

AIM:

To compute the GCD of two numbers

Algorithm:

1. Read two input values using input function
2. Convert them into integers
3. Define a function to compute GCD
 - a. Find smallest among two inputs
 - b. Set the smallest
 - c. Divide both inputs by the numbers from 1 to smallest+1If the remainders of both divisions are zero
Assign that number to gcd
- d. Return the gcd
4. Call the function with two inputs
5. Display the result

Program :

```
defcomputeGCD (x, y) :  
    if x < y :  
        smaller = x  
    else:  
        smaller = y  
    for i in range(1, smaller+1) :  
        if (x % i == 0) and (y % i == 0) :  
            gcd = i  
    returngcd  
  
num1 = int(input("Enter first number: "))  
num2 = int(input("Enter second number: "))  
print("The G.C.D. of ", num1," and ", num2," is = ", computeGCD(num1, num2))
```

Output:

```
Enter first value: 12  
Enter second value: 20  
The GCD of 12 and 20 is = : 4
```

Result:

Thus, the program to compute the GCD of two numbers has been executed successfully.

Ex. No2 FIND THE SQUARE ROOT OF A NUMBER (NEWTON'S METHOD)

Aim:

To find the square root of a number (Newton's method)

Algorithm:

1. Read one input value using input function
2. Convert it into float
3. Find the square root of the given number using the formula $\text{inputvalue}^{** 0.5}$
4. Print the result
5. Exit.

Program :

```
A=input("enter a value")  
x=float(A)  
C=x**0.5  
print("square root of ", x , "is =",C)
```

Output:

```
C:\Python36-32\mypgm>ex2.py  
Enter a value:16  
Square root of 16 is= 4.0
```

Result:

Thus, the program to find the square root of a number (Newton's method) has been executed Successfully.

Ex. No3

EXPONENTIATION (POWER OF A NUMBER)

Aim:

To write a python program to find exponentiation (power of a Number)

Algorithm:

1. Create a function using def

2. Check if exponent value

a. If exponent == 1

Return base value

b. else

Recursively call the function with (base, exponent-1)

Multiply the result returned by each recursive call with base value and Return the final result

3. Read base & exponent values using input function

4. Call the function & print the Result

Program:

```
def power(base,exp):  
    if exp==1:  
        return(base)  
    if exp!=1:  
        return(base*power(base,exp-1))  
base=int(input("Enter base: "))  
exp=int(input("Enter exponential value: "))  
print("Result:",power(base,exp))
```

Output:

```
Enter base: 2  
Enter exponential value: 2  
Result: 4
```

Result:

Thus, the program to find exponentiation (power of a Number) is executed successfully

Ex. No4**FIND THE MAXIMUM OF A LIST OF NUMBERS****Aim:**

To write a python program to find Maximum of a List of Numbers

Algorithm:

1. Initialize a List
2. Read the Number of Elements
3. Read the List values one by one
4. Set Max = first element of the list
5. Compare Max with List elements
6. If $\text{Max} < \text{List element}$
7. Set $\text{Max} = \text{List element}$
8. Continue the step 5 until reach the end of the List
9. Print the Max

Program :

```
defMaxList():
    list1=[]
    N=int(input("Enter Number of Values"))
    for i in range (0,N):
        x=int(input("Enter a Value"))
        list1.append(x)
    print(list1)
    Max=list1[0]
    j=1
    while j<N:
        if list1[j]>Max :
            Max=list1[j]
        j=j+1
```

```
print(" The Maximum Element in the List is ",Max)  
MaxList()
```

Output:

```
Enter Number of Values: 3  
Enter a Value56  
Enter a Value76  
Enter a Value43  
[56, 76, 43]  
The Maximum Element in the List is76
```

Result:

Thus, the program to find Maximum of a List of Numbers is executed successfully

Aim:

To write a python program to perform Linear Search

Algorithm:

1. Define a function called LS()
2. Define a empty list Set a flag to 0
3. Read number of elements, store it in a variable called N
4. Read the values one by one using for loop
 - a. Append each value into the List
5. Print the list
6. Read key values to be searched in the list
7. Check whether key present in the list one by one using loop statement
 - a. if found ,Set flag to 1
8. If flag==1 , then print “key is found at position *pos_value*”
9. Else , print “key not found”
10. Call the function LS() at end of code

Program:

```
def LS() :  
    list1=[]  
    flag=0  
    N=int(input("enter no. of elements"))  
    print("Enter values one by one")  
    for i in range(0,N):  
        a=int(input("enter a value"))  
        list1.append(a)  
    print(list1)  
    key=int(input("enter the Key value"))  
    for i in range(0,N) :  
        if key == list1[i]:  
            flag=1  
            break  
    if flag==1:  
        print(key ," is found in the list1 at position ",i+1 )  
    else:  
        print(key ," is not found in the list1")  
LS()
```

Output:

Enter no.of values4 Enter values one
by one enter value11
enter value22 enter value33
enter value44 [11, 22, 33, 44]
enter key value22
22 is present in the List at position 2

Result:

Thus, the Linear Search has been performed successfully

Aim:

To write a python program to perform selection sort

Algorithm:

1. Define a function selection()
2. Call this function by passing a list as input to this function
3. For every element in the list
 - a. Pick the first element
 - b. Compare it with the remaining elements of the list
 - c. Find an element that is smaller than the first one
 - d. Swap these two elements (smallest,first)
 - e. Now, the smallest element placed at first position, Do the step 3 for next element
 - f. Continue the step3 until all the elements are arranged in sorted order
4. Return the sorted list
5. Display the sorted list

Program:

```
def selection(list2):
    for i in range(len(list2)):
        least=i
        for k in range(i+1,len(list2)):
            if list2[k] <list2[least]:
                least=k
        list2=swap (list2,least,i)
    return list2

def swap(A,x,y):
    tmp=A[x]
    A[x]=A[y]
    A[y]=tmp
    return A

list1=[25,9,8,3,5,7,10]
print("Before Sorting " ,list1)
result=selection(list1)
print("After Sorting " ,result)
```

Output:

BeforeSorting [25, 9, 8, 3, 5, 7,10]

AfterSorting [3, 5, 7, 8, 9, 10,25]

Result:

Thus, the program for selection sort has been executed successfully

Ex.No 7

INSERTION SORT

Aim:

To write a python program to perform Insertion sort

Algorithm:

1. Define a function Insertion()
 - a. For every element in the list
 - i. Set current element
 - ii. Compare the current element with its previous adjacent element
If current element < previous adjacent element
Swap these two elements
Continue the step (ii) until all the predecessors to the current element are arranged in order
 - iii. Repeat the step 1a until all elements in the list are arranged in sorted order
 - b. Print the sorted list
2. Get a List
3. Call the function by passing the list as input
4. Display the result

Program:

```
def insertion(a):
    for i in a :
        j = a.index(i)
        while j>0 :
            if a[j-1] > a[j] :
                a[j-1],a[j] = a[j],a[j-1]
            else:
                break
            j = j-1

    print("After Sorting : ",a)

list1 = [16,19,11,15,10,12,14,5]
print("Before Sorting :",list1)
insertion(list1)
```

Output:

Before Sorting: [16, 19, 11, 15, 10, 12, 14, 5]

After Sorting: [5, 10, 11, 12, 14, 15, 16, 19]

Result:

Thus, the program to find first N prime Numbers has been executed successfully

Ex. No8

FIRST N PRIMENUMBERS

Aim:

To write a python program to find first N prime Numbers

Algorithm:

1. Read lower & Upper bound values for the Range
2. For each number in the Range
3. Divide the num by 2 to num-1
4. Check the remainder
 - If Remainder == 0 then,
Num is not a prime number.
 - Else
Print the Number as Prime
5. Repeat the Steps 2 to 4 until reach the Upper bound of the Range

Program :

```
lower = int(input("Enter lower range: "))
upper = int(input("Enter upper range: "))
print("Prime numbers between",lower,"and",upper,"are:")
for num in range(lower,upper + 1):
    if num > 1:
        for i in range(2,num):
            if (num%i) == 0:
                break
        else:
            print(num)
```

Output:

```
Enter lower range: 2
Enter upper range: 10
Prime numbers between 1 and 10 are:
2
3
5
7
```

Result:

Thus, the program to find first N prime Numbers has been executed successfully

Ex. No9

MULTIPLICATION OF TWOMATRICES

Aim:

To write a python program to multiply two matrices

Algorithm:

1. Create two lists with nested index
2. Initialize an empty list
3. Multiply two matrices
4. Store the results into empty list
5. Display the result

Program:

```
matrix1 = [[1, 2, 3],[1, 2, 3],[1, 2, 3]]
matrix2 = [[1, 1, 1, 1], [1, 1, 1, 1],[1, 1, 1, 1]]
rmatrix = [[0, 0, 0, 0],[0, 0, 0, 0],[0, 0, 0, 0]]
for i in range(len(matrix1)):
    for j in range(len(matrix2[0])):
        for k in range(len(matrix2)):
            rmatrix[i][j] += matrix1[i][k] * matrix2[k][j]

for r in rmatrix:
    print(r)
```

Output:

```
[6, 6, 6, 6]
[6, 6, 6, 6]
[6, 6, 6, 6]
```

Result:

Thus, the program to multiply two matrices has been executed successfully

Ex. No10**FIND THE MOST FREQUENT WORDS IN A TEXT READ FROM A FILE****Aim:**

Write a python program to find the most frequent words in a text read from a file

Algorithm:

1. Get the input file to be read.
2. Initialize num_words to zero.
3. Open the input file in read mode.
4. Use for loop to check the blank space, if there is a space then split the words and counts the words.
5. Repeat the step 4 until reach end of file.
6. Print the number of words.

Program:

```
fname = input("Enter file name: ")
num_words = 0
with open(fname, 'r') as f:
    for line in f:
        words = line.split()
        num_words += len(words)
print("Number of words:")
print(num_words)
```

To create a text file : file->new->type sentences->save it as data1.txt

Hai Welcome to all

Output:

Enter file name: data1.txt

Number of words:

4

Result:

Thus, the program to find the most frequent words in a text read from a file has been executed successfully