|  |  |  |
| --- | --- | --- |
| **Name: Naren Chandrashekhar** | **SRN: PES2UG20CS216** | **Section: G** |
| **Date: 27/05/2021** | **Week Number: 4** |

|  |  |
| --- | --- |
| **1** | Write a function to display an array elements in the reverse order using multiple files.  a) using index  b) using pointer  **Input:**  Enter the size of an array  5  Enter elements  11  22  33  44  55  **Output:**  Array elements:  11 22 33 44 55  Reversed array:  55 44 33 22 11 |
|  | **Program:**  **CLIENT FILE- Program1.c**  **#include<stdio.h>**  **#include "reverse.h"**  **int main()**  **{**  **int a[100],i,n;**  **printf("Enter the number of elements ");**  **scanf("%d",&n);**  **printf("Enter the elements ");**  **for(i=0;i<n;i++)**  **{**  **scanf("%d",&a[i]);**  **}**  **printf("The array is ");**  **for(i=0;i<n;i++)**  **{**  **printf("%d ",a[i]);**  **}**  **printf("\nThe reversed array using index is ");**  **reverse(a,n);**  **printf("\nThe reversed array using pointers is ");**  **reverse1(a,n);**  **return 0;**  **}**  **reverse.h**  **void reverse(int a[], int n);**  **void reverse1(int \*a, int n);**  **reverse.c**  **#include <stdio.h>**  **void reverse(int a[], int n)**  **{**  **int i;**  **for(i=n-1;i>=0;i--)**  **{**  **printf("%d ",a[i]);**  **}**  **}**  **void reverse1(int \*a, int n)**  **{**  **int i;**  **for(i=n-1;i>=0;i--)**  **{**  **printf("%d ",\*(a+i));**  **}**  **}** |
|  | **Output Screenshot:** |
| **2** | Write a function for factorial using recursion and use it to find C(n, r) using multiple files.  **Input:**  5 2  **Output:**  ncr is: 10 |
|  | **Program:**  **CLIENT FILE – Program2.c**  **#include<stdio.h>**  **#include "factorial.h"**  **int main()**  **{**  **int num,factorial,n,r,comb;**  **printf("Enter a number ");**  **scanf("%d",&num);**  **factorial = fact(num);**  **printf("The factorial of %d is %d ",num,factorial);**  **printf("\nEnter n and r for C(n,r) in the respective order ");**  **scanf("%d %d",&n,&r);**  **comb = combination(n,r);**  **printf("The combination is %d ",comb);**  **return 0;**  **}**  **factorial.h**  **int fact(int num);**  **int combination(int n, int c);**  **factorial.c**  **int fact(int n)**  **{**  **if(n==0)**  **return 1;**  **else**  **return n\*fact(n-1);**  **}**  **int combination(int n, int r)**  **{**  **return fact(n)/(fact(r)\*(fact(n-r)));**  **}** |
|  | **Output Screenshot:** |
| **3** | Write a C program to print all unique elements of an array using Make file  **Input:**  Input the number of elements to be stored in the array: 5  Input 5 elements in the array :  element - 0 : 1  element - 1 : 2  element - 2 : 1  element - 3 : 3  element - 4 : 3  **Output:**  The unique elements found in the array are:  List of Unique Array Elements in this Array are : 2 |
|  | **Program:**  **CLIENT FILE – Program3.c**  **#include<stdio.h>**  **#include "unique.h"**  **int main()**  **{**  **int a[100],i,n;**  **printf("Enter the number of elements ");**  **scanf("%d",&n);**  **printf("Enter the elements ");**  **for(i=0;i<n;i++)**  **scanf("%d",&a[i]);**  **unique(a,n);**  **return 0;**  **}**  **unique.c**  **#include<stdio.h>**  **int unique(int a[], int n)**  **{**  **int b[100],i,count,j,flag;**  **b[0]=a[0];**  **count = 1;**  **for(i=1;i<n;i++)**  **{**  **flag = 0;**  **for(j=0;j<count;j++)**  **{**  **if(a[i]!=b[j])**  **continue;**  **else**  **{**  **flag = 1;**  **break;**  **}**  **}**  **if(!flag)**  **{**  **b[count] = a[i];**  **count++;**  **}**  **}**  **printf("The unique elements in the array is ");**  **for(j=0;j<count;j++)**  **printf("%d ",b[j]);**    **}**  **unique.h**  **int unique(int a[], int n);**    **makefile**  **arrayunique : Program3.o unique.o**  **gcc -o arrayunique Program3.o unique.o**  **Program3.o : Program3.c unique.h**  **gcc -c Program3.c**  **unique.o : unique.c unique.h**  **gcc -c unique.c** |
|  | **Output Screenshot:** |
| **4** | Write a C program to Calculate the power of any number using recursion and multiple files  **Input:**  Recursion : Calculate the power of any number :  Input the base value : 4  Input the value of power : 2  **Output:**  The value of 4 to the power of 2 is : 16 |
|  | **Program:**  **CLIENT FILE – Program4**  **#include<stdio.h>**  **#include "p.h"**  **int main()**  **{**  **int base,pow,p;**  **printf("Enter the base and power ");**  **scanf("%d %d",&base,&pow);**  **p = power(base,pow);**  **printf("The power of %d^%d is %d ",base,pow,p);**  **return 0;**  **}**  **power1.c**  **int power(int base, int pow)**  **{**  **if(base==0)**  **return 0;**  **else if(pow==0)**  **return 1;**  **else**  **return base\*power(base,pow-1);**  **}**  **power.h**  **int power(int base, int pow);** |
|  | **Output Screenshot:** |
| **5** | Write a function to check whether a given number is prime and use that to find the next prime number, greater than a given number.  **Input1:**  Enter a number  4  **Output1:**  Next prime number=5  **Input2:**  Enter a number  113  **Ouput2:**  Next prime number=127 |
|  | **Program:**  **#include<stdio.h>**  **#include<conio.h>**  **int isPrime(int num);**  **int nextPrime(int num);**  **int main()**  **{**  **int num,np;**  **printf("Enter a number to check prime or not ");**  **scanf("%d",&num);**  **if(isPrime(num))**  **printf("The number %d is a prime ",num);**  **else**  **printf("The number %d is not a prime ",num);**  **np = nextPrime(num);**  **printf("\nThe next prime number is %d",np);**  **return 0;**  **}**  **int isPrime(int num)**  **{**  **int i;**  **for(i=2;i<num;i++)**  **{**  **if(num%i==0)**  **return 0;**  **}**  **return 1;**  **}**  **int nextPrime(int num)**  **{**  **num = num+1;**  **while(1)**  **{**  **if(isPrime(num))**  **return num;**  **num++;**  **}**  **}** |
|  | **Output Screenshot:** |
| **1** | **Practice Programs**  Write a program in C to find the maximum and minimum element in an array  **Input:**  Find maximum and minimum element in an array :  Input the number of elements to be stored in the array :5  Input 5 elements in the array :  element - 0 : 12  element - 1 : 10  element - 2 : 6  element - 3 : 7  element - 4 : 56  **Output:**  Maximum element is : 56  Minimum element is : 6 |
|  | **Program:**  **#include<stdio.h>**  **#include<conio.h>**  **int maximum(int a[],int n);**  **int minimum(int a[],int n);**  **int main()**  **{**  **int min,max,a[50],i,n;**  **printf("Enter the number of elements ");**  **scanf("%d",&n);**  **printf("Enter the elements ");**  **for(i=0;i<n;i++)**  **scanf("%d",&a[i]);**  **max = maximum(a,n);**  **min = minimum(a,n);**  **printf("The maximum and minimum of the given array is %d and %d respectively",max,min);**  **return 0;**  **}**  **int maximum(int a[],int n)**  **{**  **int i,max;**  **max = a[0];**  **for(i=1;i<n;i++)**  **{**  **if(a[i]>max)**  **max = a[i];**  **}**  **return max;**  **}**  **int minimum(int a[],int n)**  **{**  **int i,min;**  **min = a[0];**  **for(i=1;i<n;i++)**  **{**  **if(a[i]<min)**  **min = a[i];**  **}**  **return min;**  **}** |
|  | **Output Screenshot:** |
| 2 | Write a function to populate an array with fibonacci numbers using make files  **Input:**  Enter how many Fibonacci numbers you want populate:  5  **Output:**  Fibonacci number are:  0  1  1  2  3 |
|  | **Program:**  **CLIENT FILE – PracticeProgram2.c**  **#include<stdio.h>**  **#include<conio.h>**  **#include "fib.h"**  **int main()**  **{**  **int n;**  **printf("Enter the number of terms ");**  **scanf("%d",&n);**  **fibonacci(n);**  **return 0;**  **}**  **fib.h**  **void fibonacci(int n);**  **fib.c**  **#include<stdio.h>**  **void fibonacci(int n)**  **{**  **int i,a[50];**  **a[0] = 0;**  **a[1] = 1;**  **printf("%d %d ",a[0],a[1]);**  **for(i=2;i<=n;i++)**  **{**  **a[i] = a[i-2] + a[i-1];**  **printf("%d ",a[i]);**  **}**    **}**  **Makefile**  **fibseries : PracticeProgram2.o fib.o**  **gcc -o fibseries PracticeProgram2.o fib.o**  **PracticeProgram2.o : PracticeProgram2.c fib.h**  **gcc -c PracticeProgram2.c**  **fib.o : fib.c fib.h**  **gcc -c fib.c** |
|  | **Output Screenshot:** |