|  |  |  |
| --- | --- | --- |
| **Name: Jacob V Sanoj** | **SRN: PES1UG20EC083** | **Section:**  **F1** |
| **Date: 29-05-2021** | **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:  #include <stdio.h>  #include "Question1\_reverseindex.c"  #include "Question1\_reversepointer.c"  void main()  {  printf("Enter the size of the array :\n");  int n;  scanf("%d", &n);  int array[n];  printf("Enter the elements of the array :\n");  for(int i = 0; i < n; i++)  scanf("%d", &array[i]);  reverse\_index(n, array);  printf("\n");  reversearray(array, n);  }  #include <stdio.h>  int reverse\_index(int n, int array[])  {  printf("Array elements :\n");  for(int i = 0; i < n; i++)  printf("%d ", array[i]);  printf("\n");  printf("Reversed elements :\n");  for(int i = n-1; i >= 0; --i)  printf("%d ", array[i]);  }  #include <stdio.h>  void reversearray(int \*p, int n)  {  int \*first = p;  int \*last = p+n-1;  while(first<last)  {  int temp = \*first;  \*first = \*last;  \*last = temp;  first++;  last--;  }  printf("Reversed array elements using pointers are: \n");  for(int i=0; i<n; i++)  printf("%d ", \*p++);  printf("\n");  } |
|  | **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:  #include <stdio.h>  #include "Question\_2\_client2.c"  int factorial(int n);  int main()  {  printf("Enter the value of n and r in nCr \n");  int n, r;  scanf("%d%d", &n, &r);  int answer = C(n, r);  printf("The value of nCr is %d\n", answer);  }  #include <stdio.h>  #include "Question\_2\_client.c"  int C(int n, int r)  {  return factorial(n)/(factorial(n-r) \* factorial(r));  }  #include <stdio.h>  int factorial(int n)  {  if(n == 0)  return 0;  if (n == 1)  return 1;  else  return n \* factorial(n-1);  } |
|  | **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:  #include <stdio.h>  #include "Question3\_uniqueElement.c"  int main()  {  int a[100], n;  printf("Enter the number of elements in the array\n");  scanf("%d", &n);  printf("Enter the elements\n");  int i = 0;  while (i < n)  {  scanf("%d", &a[i]);  i++;  }  uniqueElement(a, n);  } |
|  | **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:  #include <stdio.h>  #include "Question3\_power.c"  int main()  {  int n, pow;  printf("Enter the number and the power that you want to calculate for :\n");  scanf("%d%d", &n, &pow);  int ans = ppow(n, pow);  printf("The power of %d to %d is : %d\n", n, pow, ans);  }  #include <stdio.h>  int ppow(int num, int pow)  {  if(pow == 0)  return 1;  else  return num \* ppow(num, pow-1);  } |
|  | **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 |
|  | #include <stdio.h>  int prime(int n);  int main()  {  int n;  printf("Enter the number that you want to check if it is prime: \n");  scanf("%d", &n);  int count = prime(++n);  while (count != 0)  count = prime(++n);  printf("The next prime number is %d\n", n);  }  int prime(int n)  {  int count = 0;  if (n == 0 || n == 1)  {  printf("It is neither a prime nor composite\n");  return 2;  }  else  for (int i = 2; i < n; i++)  if (n % i == 0)  {  count = 1;  break;  }  return count;  } |
|  | **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>  int main()  {  int arr[100], n, max = 0, min = 0;  printf("Enter the number of elements in the array :\n");  scanf("%d", &n);  printf("Enter the values of the array :\n");  for(int i = 0; i < n; i++)  scanf("%d", &arr[i]);  max = arr[0];  min = arr[0];  for(int i = 0; i < n; i++)  {  if(arr[i] > max)  max = arr[i];  if(arr[i] < min)  min = arr[i];  }  printf("The Maximum element is %d\nThe Minimum element is %d\n", max, 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:  #include <stdio.h>  void fib(int n)  {  int a = 0, b = 1;  int term = a + b;  printf("Fibonacci Series: \n%d \n%d \n", a, b);  while ((n - 1) != 0)  {  printf("%d \n", term);  a = b;  b = term;  term = a + b;  n--;  }  }  #include <stdio.h>  #include "QuestionPractice\_2\_fibonacci.c"  int main()  {  int n;  printf("Enter the range: ");  scanf("%d", &n);  fib(n);  } |
|  | **Output Screenshot:** |