|  |  |  |
| --- | --- | --- |
| **Name:**  **P K Navin Shrinivas** | **SRN:PES2UG20CS237** | **Section:G(B-2)** |
| **Date:27/05/21** | **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:**  **1\_1.c:**  #include<stdio.h>  #include<stdbool.h>  #include<math.h>  void indexreverse(int a[],int n){  printf("Reversed array[Using Index]:");  for(int i=0;i<n;i++)  {  printf("%i ",a[n-i-1]);  }  printf("\n");  }  void pointerreverse(int a[],int n){  int\* right=(a+n-1);  printf("Reversed array[Using Pointer]:");  for(int i=0;i<n;i++)  {  printf("%i ",\*(right-i));  }  printf("\n");  }  **1\_1.h:**  void indexreverse(int a[],int n);  void pointerreverse(int a[],int n);  **1prog.c:**  #include<stdio.h>  #include<stdbool.h>  #include<math.h>  #include"1\_1.h"  int main(){  int n;  printf("Enter number of elements : ");  scanf("%i",&n);  int a[n];  printf("Enter the elements :");  for(int i=0;i<n;i++)  scanf("%i",a+i);  pointerreverse(a,n);  indexreverse(a,n);  } |
|  | **Output Screenshot:**  **1** |
| **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:**  **2\_1.c:**  #include<stdio.h>  int factorial(int n){  if(n==1)  return 1;  return n\*factorial(n-1);  }  int ncr(int n,int r)  {  if(n<r)  {  printf("Invalid values!");  return 0;  }  return factorial(n)/(factorial(r)\*factorial(n-r));  }  **2.1.h:**  int factorial(int n);  int ncr(int n,int r);  **2prog.c:**  #include"2\_1.h"  #include<stdio.h>  int main(){  int n,r;  printf("Enter two number in the form n r to find combinations :");  scanf("%i %i",&n,&r);  if(ncr(n,r)!=0)  printf("nCr(%i,%i) = %i",n,r,ncr(n,r));  } |
|  | **Output Screenshot:**  **2** |
| **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:**  **3prog.c:**  //this solution is a O(k)solution compared to the O(n^2) basic solution  //where k is the largest element value  #include<stdio.h>  int main(){  int n;  printf("Enter number of elements :");  scanf("%i",&n);  int a[n];  int largest=-2147483647;  printf("Enter the elements :\n ");  for(int i=0;i<n;i++)  {  printf("\t Element %i : ",i+1);  scanf("%i",a+i);  if(a[i]>largest)  largest=a[i];  }  int fakemap[largest+1];  for(int i=0;i<=largest;i++)  {  fakemap[i]=0;  }  for(int i=0;i<n;i++)  {  fakemap[\*(a+i)]++;  }  printf("Unique element(s) in the array :");  for(int i=0;i<=largest;i++)  {  if(fakemap[i]==1)  printf("%i ",i);  }  printf("\n");  }  **3make.mk:**  3prog.o : 3prog.c  gcc 3prog.c -o 3prog.o |
|  | **Output Screenshot:**  **3** |
| **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:**  **4\_1.c:**  int power(int b,int p)  {  if(p==0)  return 1;  return b\*power(b,p-1);  }  **4\_1.h:**  int power(int b,int p);  **4prog.c:**  #include<stdio.h>  #include "4\_1.h"  int main(){  int b,p;  printf("Enter base and power : ");  scanf("%i %i",&b,&p);  printf("Value of %i to the power %i is : %i",b,p,power(b,p));  } |
|  | **Output Screenshot:**  **4** |
| **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:**  **5\_1.c:**  *#include*<stdbool.h>  *bool* *isprime*(*int* n)  {  *for*(*int* i*=*2;i*\**i*<=*n*+*1;i*++*)  {  *if*(n*%*i*==*0)  *return* false;  }  *return* true;  }  *int* *nextprime*(*int* n)  {  *int* c*=*n*+*1;  *while*(*isprime*(c)*==*false)  c*++*;  *return* c;  }  **5\_1.h:**  bool isprime(int n);  int nextprime(int n);  **5prog.c:**  #include<stdio.h>  #include<stdbool.h>  #include"5\_1.h"  int main()  {  int n;  printf("Enter a number : ");  scanf("%i",&n);  if(isprime(n))  printf("Yes!%i is a prime \n",n);  else  printf("No!%i is not a prime \n",n);  printf("Next prime number found is : %i",nextprime(n));  } |
|  | **Output Screenshot:**  **5** |
| **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:**  **prog1.c:**  *#include*<stdio.h>  *//one does not need an array do this prgram but as the question mentions*  *//arrays multiple times i am using one*  *int* *main*(){  *int* largest*=-*2147483647;  *int* smallest*=*2147483647;  *int* si*=*0,li*=*0,n;  *printf*("Enter number of elements : ");  *scanf*("%i",*&*n);  *int* *a*[n];  *for*(*int* i*=*0;i*<*n;i*++*)  {  *printf*("Element %i : ",i*+*1);  *scanf*("%i",a*+*i);  *if*(*a*[i]*>*largest)  {  largest*=a*[i];  li*=*i;  }  *if*(*a*[i]*<*smallest)  {  smallest*=a*[i];  si*=*i;  }  }  *printf*("Largest element is : %i \n",*a*[li]);  *printf*("Smallest element is : %i",*a*[si]);  } |
|  | **Output Screenshot:**  **6** |
|  |  |
| 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:**  **7make.mk:**  all : 7\_1.o 7prog.o  7\_1.o : 7\_1.c  gcc -c 7\_1.c  7prog.o : 7\_1.o 7prog.c  gcc 7\_1.o 7prog.c -o 7prog.o  clean:  rm -r 7prog.o; \  rm -r 7\_1.o  **7\_1.c:**  *void* *populatefibbonaci*(*int* a*[]*,*int* n)  {  *int* first*=*0;  *int* second*=*1;  *if*(n*<=*2){  *a*[0]*=*first;  *a*[1]*=*second;  }  *else*{  *a*[0]*=*first;  *a*[1]*=*second;  *for*(*int* i*=*2;i*<*n;i*++*)  {  *a*[i]*=*first*+*second;  first*=*second;  second*=a*[i];  }  }  }  **7\_1.h:**  void populatefibbonaci(int a[],int n);  **7prog.c:**  #include<stdio.h>  #include"7\_1.h"  int main(){  int n;  printf("Enter number of fibbonaci to populate : ");  scanf("%i",&n);  int a[n];  populatefibbonaci(a,n);  printf("Generated fibbonaci series is : \n");  for(int i=0;i<n;i++)  {  printf("%i ",a[i]);  }  } |
|  | **Output Screenshot:**  **7** |