|  |  |  |
| --- | --- | --- |
| **Name: Renita Kurian** | **SRN: PES1UG20CS331** | **Section: N** |
| **Date:30/05/2021** | **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** |
|  | **Program:**  **A) using index**  **P1a\_client.c:**  #include<stdio.h>  #include"P1a\_server.h"  int main()  {  int a[100];int n;int i;  printf("Enter the size of an array\n");  scanf("%d",&n);  read\_array(a,n);  printf("Array elements: \n");  disp\_array(a,n);  rev\_array(a,0,n-1);  printf("Reversed array: \n");  disp\_array(a,n);  return 0;  }  **P1a\_server.c:**  #include<stdio.h>  #include"P1a\_server.h"  void rev\_array(int a[],int start,int end)  {  int temp;  while(start<end)  {  temp=a[start];  a[start]=a[end];  a[end]=temp;  ++start;  --end;  }}  void read\_array(int a[],int n)  {  int i;  printf("Enter elements\n");  for(i=0;i<n;++i)  scanf("%d",&a[i]);  }  void disp\_array(int a[],int n)  {  int i;  for(i=0;i<n;++i)  printf("%d\t",a[i]);  printf("\n");  }  **P1a\_server.h:**  void rev\_array(int[],int ,int);  void read\_array(int[],int);  void disp\_array(int[],int);  **B) using pointer**  **P1b\_client.c:**  #include<stdio.h>  #include"P1b\_server.h"  int main()  {  int a[100];int n;int i;  printf("Enter the size of an array\n");  scanf("%d",&n);  read\_array(a,n);  printf("Array elements: \n");  disp\_array(a,n);  rev\_array(a,0,n-1);  printf("Reversed array: \n");  disp\_array(a,n);  return 0;  }  **P1b\_server.c:**  #include<stdio.h>  #include"P1b\_server.h"  void rev\_array(int a[],int start,int end)  {  int temp;  while(start<end)  {  temp=\*(a+start);  \*(a+start)=\*(a+end);  \*(a+end)=temp;  ++start;  --end;  }}  void read\_array(int a[],int n)  {  int i;  printf("Enter elements\n");  for(i=0;i<n;++i)  scanf("%d",&a[i]);  }  void disp\_array(int a[],int n)  {  int i;  for(i=0;i<n;++i)  printf("%d\t",a[i]);  printf("\n");  }  **P1b\_server.h:**  void rev\_array(int[],int ,int);  void read\_array(int[],int);  void disp\_array(int[],int); |
|  | **Output Screenshot:**  **a)**    **b)** |
| **2** | Write a function for factorial using recursion and use it to find C(n,r) using multiple files |
|  | **Program:**  **P2\_client.c:**  #include<stdio.h>  #include"P2\_server.h"  int main()  {  int n; int r; int ncr;  printf("Enter the value of n and r\n");  scanf("%d %d",&n,&r);  ncr=fact(n)/(fact(r)\*fact(n-r));  printf("ncr is: %d",ncr);  return 0;  }  **P2\_server.c:**  #include"P2\_server.h"  int fact(int n)  {  return(n==0)?1:n\*fact(n-1);  }  **P2\_server.h:**  int fact(int); |
|  | **Output Screenshot:** |
| **4** | Write a program in C to calculate the power of any number using recursion and multiple files |
|  | **Program:**  **P4\_client.c:**  #include<stdio.h>  #include"P4\_server.h"  int main()  {  int bNum;int pwr;  long int result;  printf("Input the base value: ");  scanf("%d",&bNum);  printf("Input the value of power : ");  scanf("%d",&pwr);  result=Power(bNum,pwr);  printf("The value of %d to the power of % is: %d\n\n",bNum,pwr,result);  return 0;  }  **P4\_server.c:**  #include<stdio.h>  #include"power.h"  long int Power(int x,int y)  {  long int result=1;  if(y==0)  return result;  else  result=x\*(Power(x,y-1));  }  **P4\_server.h:**  long int Power(int x,int y); |
|  | **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 |
|  | **Program:**  **P5\_client.c:**  #include<stdio.h>  #include"P5\_server.h"  int main()  {  int n;  printf("Enter a number\n");  scanf("%d",&n);  printf("Next prime number=%d\n",nextprime(n));  return 0;  }  **P5\_server.c:**  #include"server.h"  int isprime(int n)  {  int i,count=0;  for(i=1;i<=n;++i)  {  if(n%i==0)  count++;  }  if(count==2)  return 1;  else if(count>2)  return 0;  }  int nextprime(int n)  {  int i=n+1;  while(1)  {  if(isprime(i))  break;  i++;  }  return i;  }  **P5\_server.h:**  int isprime(int n);  int nextprime(int n); |
|  | **Output Screenshot:** |