**Array Implementation**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int choice,arr[20],i,j,k,isize,dsize,sesize,ipos,dpos,num,findElement,found;

printf("Array Implementation\n");

printf("--------------------\n");

printf("| 1.Insertion |\n");

printf("| 2.Deletion |\n");

printf("| 3.Search |\n");

printf("| 4.Exit |\n");

printf("--------------------\n");

while(1)

{

printf("\n\nEnter Your Choice: ");

scanf("%d",&choice);

switch(choice)

{

case 1:

printf("\n---Insertion Implementation---\n\n");

printf("Enter the Size of the Array: ");

scanf("%d", &isize);

printf("\nEnter %d Elements of the Array:\n",isize);

for(i=0; i<isize; i++)

{

scanf("%d", &arr[i]);

}

printf("\nEnter the Element to Insert: ");

scanf("%d", &num);

printf("\nEnter the Element to Position: ");

scanf("%d", &ipos);

if(ipos > isize+1 || ipos <= 0)

{

printf("Invalid Position! Please enter Position between 1 to %d", isize);

}

else

{

for(i=isize; i>=ipos; i--)

{

arr[i] = arr[i-1];

}

arr[ipos-1] = num;

isize++;

printf("\n---INSERTION IMPLEMENTATION OUTPUT---\n\n");

printf("The Array Elements After Insertion: ");

for(i=0; i<isize; i++)

{

printf("%d ", arr[i]);

}

}

break;

case 2:

printf("\n\n---Deletion Implementation---\n\n");

printf("Enter the Size of the Array: \n");

scanf("%d", &dsize);

printf("\nEnter %d Elements of the Array: \n", dsize);

for(j = 0; j < dsize; j++)

scanf("%d", &arr[j]);

printf("\nEnter the Position to Delete Element: ");

scanf("%d", &dpos);

if (dpos >= dsize+1)

printf("\nDeletion not possible.\n");

else

{

for(j = dpos - 1; j < dsize - 1; j++)

arr[j] = arr[j+1];

printf("\n---DELETION IMPLEMENTATION OUTPUT---\n\n");

printf("The Array Elements After Deletion: ");

for(j = 0; j < dsize - 1; j++)

printf("%d ", arr[j]);

}

break;

case 3:

printf("\n---Search Implementation---\n\n");

printf("Enter the Size of the Array:\n");

scanf("%d", &sesize);

printf("\nEnter %d Elements of the Array:\n",sesize);

for(k=0; k<sesize; k++)

{

scanf("%d", &arr[k]);

}

printf("\nEnter the Element to Search: ");

scanf("%d", &findElement);

found = 0;

for(k=0; k<sesize; k++)

{

if(arr[k] == findElement)

{

found = 1;

break;

}

}

if(found == 1)

{

printf("\n---SEARCH IMPLEMENTATION OUTPUT---\n\n");

printf("%d is Found at Position %d\n", findElement, k + 1);

}

else

{

printf("\n---SEARCH IMPLEMENTATION OUTPUT---\n\n");

printf("%d is not Found in the Array\n", findElement);

}

break;

case 4:

printf("\nProgram Exit Successfully.\n");

exit(1);

default:

printf("Wrong Choice!\n\n");

}

}

}

**Stack Implementation**

|  |  |
| --- | --- |
| #include<stdio.h>  #include<stdlib.h>  #define MAXSIZE 5  int main()  {  struct stack{  int stk[MAXSIZE];  int top;  };  typedef struct stack STACK;  STACK s;  int choice,i,num,top;  int option = 1;  s.top = -1;  while (option)  {  printf ("Stack Implementation\n");  printf ("1.Push\n");  printf ("2.Pop\n");  printf ("3.Display\n");  printf ("4.Exit\n");  printf("\nEnter your Choice:");  scanf("%d", &choice);  switch (choice)  {  case 1:  if (s.top == (MAXSIZE - 1))  {  printf ("Stack is Full\n\n");  }  else  {  printf("Enter the Element to be Pushed: ");  scanf ("%d", &num);  s.top = s.top + 1;  s.stk[s.top] = num;  }  break; | case 2:  if (s.top == - 1)  {  printf ("Stack is Empty\n\n");  }  else  {  num = s.stk[s.top];  printf ("Poped Element is = %d\n\n", s.stk[s.top]);  s.top = s.top - 1;  }  break;  case 3:  if (s.top == -1)  {  printf ("Stack is empty\n\n");  }  else  {  printf ("\nThe Elements of the Stack are: \n");  for (i = s.top; i >= 0; i--)  {  printf("%d\t", s.stk[i]);  }  }  printf("\n");  break;  case 4:  printf("Program Exit\n");  exit(1);  break;  }  }  return 0;  } |

**Queue Implementation**

|  |  |
| --- | --- |
| #include<stdio.h>  #include<stdbool.h>  int Size = 5,Head=0,Tail=0;  int Queue[5];  int Is\_Empty()  {  if(Head==Tail)  {  printf("Queue is empty\n");  return 1;  }  return 0;  }  int Is\_Full()  {  if(Tail==Size)  {  printf("Queue is Full\n");  return 1;  }  return 0;  }  void Enqueue()  {  //printf("\n\n%d\n",Tail);  int data;  if(Is\_Full()==1)  return 0;  printf("Insert the Element in Queue:\n");  scanf("%d",&data);  Queue[Tail]=data;  Tail++;  return 1;  }  void Dequeue()  {  int data;  if(Is\_Empty()==1)  return 0;  data=Queue[Head];  Head++;  printf("%d has been De\_Queued\n",data);  return 1;  } | int Print()  {  int i;  if(Is\_Empty()==1)  return 0;  printf("The Elements are:\n");  for(i=Head;i<=Tail-1;i++)  {  printf("%d \n",Queue[i]);  }  return 1;  }  int main()  {  int x;  do{  printf("Queue Implementation\n\n");  printf("1.EnQueue\n");  printf("2.DeQueue\n");  printf("3.Print\n");  printf("4.Exit\n");  printf("\nEnter your Choice:");  scanf("%d", &x);  if(x==1)  Enqueue();  else if(x==2)  Dequeue();  else if(x==3)  Print();  }while(x!=4);  return 0;  } |

**Reverse String**

#include <stdio.h>

#include <string.h>

int main()

{

char str[100];

int i, len;

printf("Please Enter any String: ");

gets(str);

len = strlen(str);

printf("\nString after Reversing : ");

for(i = len - 1; i >= 0; i--)

{

printf("%c", str[i]);

}

printf("\n");

return 0;

}

**Palindrome**

#include <stdio.h>

int main()

{

int n, reversedInteger = 0, remainder, originalInteger;

printf("Enter an integer: ");

scanf("%d", &n);

originalInteger = n;

while(n != 0)

{

remainder = n%10;

reversedInteger = reversedInteger\*10 + remainder;

n /= 10;

}

if(originalInteger == reversedInteger)

printf("%d is a palindrome.", originalInteger);

else

printf("%d is not a palindrome.", originalInteger);

return 0;

}

**Armstrong Number**

#include <stdio.h>

int main()

{

int number, originalNumber, remainder, result = 0;

printf("Enter a three digit integer: ");

scanf("%d", &number);

originalNumber = number;

while (originalNumber != 0)

{

remainder = originalNumber%10;

result += remainder\*remainder\*remainder;

originalNumber /= 10;

}

if(result == number)

printf("%d is an Armstrong number.",number);

else

printf("%d is not an Armstrong number.",number);

return 0;

}

**Fibonacci Series and Store it in an Array**

#include <stdio.h>

int main()

{

int fib[24];

int i;

fib[0] = 0;

fib[1] = 1;

for(i = 2; i < 24; i++)

fib[i] = fib[i-1] + fib[i-2];

for(i = 0; i < 24; i++)

printf("%3d %6d\n", i, fib[i]);

}