# EXSERCISE-9

1.A

WRITE A C PROGRAM TO FIND THE SUM OF TWO 1D ARRAYS USING MALLOC()

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

void main()

{

int i, n;

int \*a, \*b, \*c;

printf("Enter the size of the arrays\n");

scanf("%d", &n);

a = (int \*)malloc(n \* sizeof(int));

b = (int \*)malloc(n \* sizeof(int));

c = (int \*)malloc(n \* sizeof(int));

printf("Enter Elements of First List\n");

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

{

scanf("%d", a + i);

}

printf("Enter Elements of Second List\n");

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

{

scanf("%d", b + i);

}

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

{

\*(c + i) = \*(a + i) + \*(b + i);

}

printf("Resultant List is\n");

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

{

printf("%d\n", \*(c + i));

}

return 0;

}

OUT PUT

Enter the size of the arrays

2

Enter Elements of First List

1

2

Enter Elements of Second List

2

2

2

Resultant List is

3

4

9.B

WRITE A C PROGRAM TO IMPLEMENT REALLOC()

#include <stdio.h>

#include <stdlib.h>

int main()

{ int i, \* ptr, sum = 0;

ptr = malloc(100);

if (ptr == NULL) {

printf("Error! memory not allocated.");

exit(0);

}

ptr = realloc(ptr,500);

if(ptr != NULL)

printf("Memory created successfully\n");

return 0;

}

OUT PUT:

Memory created successfully

EXSERCISE-10

10.A.CREATE AND DISPLAY A SINGLY LINKED LIST USING SELF –REFERENTIAL STRUCTURE

|  |
| --- |
| #include <stdio.h>    struct node {      int data1;      char data2;      struct node\* link;  };    int main()  {      struct node ob1; // Node1        // Initialization      ob1.link = NULL;      ob1.data1 = 10;      ob1.data2 = 20;        struct node ob2; // Node2        // Initialization      ob2.link = NULL;      ob2.data1 = 30;      ob2.data2 = 40;        // Linking ob1 and ob2      ob1.link = &ob2;        // Accessing data members of  ob2 using ob1      printf("%d", ob1.link->data1);      printf("\n%d", ob1.link->data2);      return 0;  } |

OUT PUT:

30

40

10.B.WRITE A C PROGRAM TO SHIFT/ROTATE USING BITFIELDS

#include <stdio.h>

#define INT\_BITS 32

/\*Function to left rotate n by d bits\*/

int leftRotate(int n, unsigned int d)

{

    /\* In n<<d, last d bits are 0. To put first 3 bits of n

       at last, do bitwise or of n<<d with n >>(INT\_BITS -

       d) \*/

    return (n << d) | (n >> (INT\_BITS - d));

}

/\*Function to right rotate n by d bits\*/

int rightRotate(int n, unsigned int d)

{

    /\* In n>>d, first d bits are 0. To put last 3 bits of at

            first, do bitwise or of n>>d with n <<(INT\_BITS

       - d) \*/

    return (n >> d) | (n << (INT\_BITS - d));

}

/\* Driver program to test above functions \*/

void main()

{

    int n = 16;

    int d = 2;

    printf("Left Rotation of %d by %d is ", n, d);

    printf("%d", leftRotate(n, d));

    printf(" Right Rotation of %d by %d is ", n, d);

    printf("%d", rightRotate(n, d));

}

**Output**

Left Rotation of 16 by 2 is 64 Right Rotation of 16 by 2 is 4

EXERCISE-11

11.A.WRITE A “C” FUNCTION TO CALCULATE NCR VALUE

#include <stdio.h>

  int fact(int z);

void main()

{

int n, r, ncr;

printf("**\n** Enter the value for N and R **\n**");

scanf("%d%d", &n, &r); ncr = fact(n) / (fact(r) \* fact(n - r));

printf("**\n** The value of ncr is: %d", ncr);

}

int fact(int z)

{

int f = 1, i;

if (z == 0)

{

return(f);

}

else

{

for (i = 1; i <= z; i++)

{

f = f \* i;

}

}

return(f);

}

OUTPUT:

Enter the value for N and R

2

7

The value of ncr is: 0

11.B.WRITE A C PROGRAM TO TRANSPOSE OF A MATRIX

#include <stdio.h>

int main() {

int a[10][10], transpose[10][10], r, c;

printf("Enter rows and columns: ");

scanf("%d %d", &r, &c);

// asssigning elements to the matrix

printf("\nEnter matrix elements:\n");

for (int i = 0; i < r; ++i)

for (int j = 0; j < c; ++j) {

printf("Enter element a%d%d: ", i + 1, j + 1);

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

}

// printing the matrix a[][]

printf("\nEntered matrix: \n");

for (int i = 0; i < r; ++i)

for (int j = 0; j < c; ++j) {

printf("%d ", a[i][j]);

if (j == c - 1)

printf("\n");

}

// computing the transpose

for (int i = 0; i < r; ++i)

for (int j = 0; j < c; ++j) {

transpose[j][i] = a[i][j];

}

// printing the transpose

printf("\nTranspose of the matrix:\n");

for (int i = 0; i < c; ++i)

for (int j = 0; j < r; ++j) {

printf("%d ", transpose[i][j]);

if (j == r - 1)

printf("\n");

}

return 0;

}

OUTPUT;

Enter rows and columns: 1 2

Enter matrix elements:

Enter element a11: 1

Enter element a12: 2

Entered matrix:

1 2

Transpose of the matrix:

1

2

EXERCISE-12

12.A.WRITE A RECURSIVE FUNCTION TO GENERATE FIBONACCI SERIES

#include<stdio.h>

void printFibonacci(int m){

static int m1=0,m2=1,m3;

if(m>0){

m3 = m1 + m2;

m1 = m2;

m2 = m3;

printf(“%d “,m3);

printFibonacci(m-1);

}

}

int main(){

int m;

printf(“Please enter your preferred number of elements here: “);

scanf(“%d”,&m);

printf(“The Fibonacci Series will be: “);

printf(“%d %d “,0,1);

printFibonacci(m-2); //We have used m-2 because we have 2 numbers already printed here

return 0;

}

OUT PUT:

Please enter your preferred number of elements here: 15

The Fibonacci Series will be:

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

12.B.WRITE A RECURSIVE FUNCTION TO FIND THE SUM OF SERIES

#include <stdio.h>

int addNumbers(int n);

int main() {

int num;

printf("Enter a positive integer: ");

scanf("%d", &num);

printf("Sum = %d", addNumbers(num));

return 0;

}

int addNumbers(int n) {

if (n != 0)

return n + addNumbers(n - 1);

else

return n;

}

OUTPUT:

Enter a positive integer: 20

Sum = 210

EXERCISE-13

13.A.WRITE A C PROGRAM TO SWAP TWO NUMBER USING CALL BY REFERENCE

#include <stdio.h>

void swap(int\*, int\*);

int main()

{

int x, y;

printf("Enter the value of x and y\n");

scanf("%d%d",&x,&y);

printf("Before Swapping\nx = %d\ny = %d\n", x, y);

swap(&x, &y);

printf("After Swapping\nx = %d\ny = %d\n", x, y);

return 0;

}

void swap(int \*a, int \*b)

{

int temp;

temp = \*b;

\*b = \*a;

\*a = temp;

}

OUTPUT:

Enter the value of x and y

2

3

Before Swapping

x = 2

y = 3

After Swapping

x = 3

y = 2

13.B.WRITE A C PROGRAM TO FIND NO OF LOWERCASE,UPPERCASE,DIGITS AND OTHER CHARACTERS USING POINTERS

#include <stdio.h>

#include <stdlib.h>

int main()

{

char str[100];

int i;

int upper=0,lower=0,num=0,special=0;;

printf("Please enter the string \n");

gets(str);

**for**(i=0; str[i] != '\0'; i++){

//check for uppercase

**if**(str[i]>='A' && str[i]<='Z') {

upper++;

}**else** **if**(str[i]>='a' && str[i]<='z') {//check lower case

lower++;

}**else** **if**(str[i]>='1' && str[i]<='9') { //check number

num++;

}

**else**{

special++;

}

}

printf("\nUpper case letters: %d",upper);

printf("\nLower case letters: %d",lower);

printf("\nNumbers: %d",num);

printf("\nSpecial characters: %d",special);

getch();

**return** 0;

}

OUT PUT:

Please enter the string

Murali Mohan 18@

Upper case letters: 2

Lower case letters: 15

Numbers:2

Special characters: 1

EXERCISE-14

14.A.WRITE A C PROGRAM TO WRITE A AND READ TEXT INTO A FILE



#include< stdio.h >

**int** main()

{

FILE \*fp; /\* file pointer\*/

**char** fName[20];

printf("\nEnter file name to create :"); scanf("%s",fName); /\*creating (open) a file\*/ fp=fopen(fName,"w"); /\*check file created or not\*/ **if**(fp==NULL)

{

printf("File does not created!!!");

exit(0); /\*exit from program\*/

}

printf("File created successfully."); /\*writting into file\*/

putc('A',fp);

putc('B',fp);

putc('C',fp);

printf("\nData written successfully.");

fclose(fp); /\*again open file to read data\*/ fp=fopen(fName,"r");

**if**(fp==NULL)

{

printf("\nCan't open file!!!");

exit(0);

}

printf("Contents of file is :\n"); printf("%c",getc(fp));

printf("%c",getc(fp));

printf("%c",getc(fp));

fclose(fp);

**return** 0;

}

Output

Enter file name to create : ok.txt File created successfully. Data written successfully. Contents of file is : ABC

14.B.WRITE A C PROGRAM TO PRINT LAST N CHARACTERS OF GIVEN FILE

#include<stdio.h>

int main()

{

FILE \*fp;

char ch; // Read last num characters from end

int number = 10;

long length;

fp = fopen("opengenus.txt", "r");

if (fp == NULL)

{

puts("cannot open this file");

exit(1);

}

fseek(fp, 0, SEEK\_END);

length = ftell(fp);

fseek(fp, (length - number), SEEK\_SET);

do

{

ch = fgetc(fp);

putchar(ch);

}

while (ch != EOF);

fclose(fp);

return(0);

}

OUT PUT:

cannot open this file