ASSIGNMENT 2 23/03/2023

Program 1

Output:

#include <stdio.h>

int main() {

int a, b, c, max;

printf("Enter three numbers: ");

scanf("%d %d %d", &a, &b, &c);

max = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

printf("Maximum number is %d", max);

return 0;

}

Program 2

Output:

#include <stdio.h>

int main()

{

char ch;

printf("Enter a character: ");

scanf("%c", &ch);

(ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z') ? printf("%c is an alphabet", ch) : printf("%c is not an alphabet", ch);

return 0;

}

Program 3

Output:

#include <stdio.h>

int main() {

int n, sum = 0;

printf("Enter the value of n: ");

scanf("%d", &n);

for(int i = 1; i <= n; i += 2)

{

sum += i;

}

printf("The sum of all odd numbers from 1 to %d is %d\n", n, sum);

return 0;

}

Program 4

Output:

#include <stdio.h>

#include <string.h>

int main()

{

char str[100];

int i, j, maxCount = 0, len;

char maxChar;

printf("Enter a string: ");

scanf("%[^\n]", str);

len = strlen(str);

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

{

int count = 0;

for(j = i+1; j < len; j++) {

if(str[i] == str[j] && str[i] != ' ')

{

count++;

}

}

if(count > maxCount) {

maxCount = count;

maxChar = str[i];

}

}

printf("Maximum occurring character is '%c' with frequency %d", maxChar,max);

return 0;

}

Program 5

Output:

#include <stdio.h>

int tmp=20;

main( ) {

 printf("%d ",tmp);

 func( );      
     printf("%d ",tmp);

}

func()

{

static int tmp=10;

 printf("%d ",tmp);

}

OUTPUT:

20 10 20

Program 6

Output:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

int main()

{

int a = 10;

pid\_t pid = fork();

if (pid == 0) {

// child process

a++;

printf("Child process: a = %d\n", a);

exit(0);

} else if (pid > 0) {

// parent process

wait(NULL);

printf("Parent process: a = %d\n", a);

} else {

// error occurred

fprintf(stderr, "fork failed\n");

exit(1);

}

return 0;

}

Program 7

Output:

#include <stdio.h>

int main () {

int a;

/\* for loop execution \*/

for( a = 1; a <= 100; a++ )

{

printf("%d\n",a \* a);

}

return 0;

}

OUTPUT:

1

4

9

16

25

36

49

64

81

100

121

144

169

196

225

256

289

324

361

400

441

484

529

576

625

676

729

784

841

Program 8

Output:

#include<stdio.h>

int main()

{

int arr[]={2, 3, 4, 1, 6};

  printf("%u, %u, %un", arr, &arr[0], &arr);

  return 0;

}

OUTPUT:

521416208, 521416208, 521416208n

Program 9

Output:

#include<stdio.h>

void fun(int \*\*p);

int main()

{

int a[3][4] = {1, 2, 3, 4, 4, 3, 2, 8, 7, 8, 9, 0};

  int \*ptr;

  ptr = &a[0][0];

  fun(&ptr);

  return 0;

}

void fun(int \*\*p)

{

    printf("%dn", \*\*p);

}

OUTPUT:

1n

Program 10

Output:

#include <stdio.h>

#include <string.h>

int main() {

char names[5][20]; // assuming we want to sort 5 names of maximum 20 characters each

char temp[20];

int i, j, n;

char order;

// Get list of names from user

printf("Enter %d names:\n", 5);

for (i = 0; i < 5; i++) {

printf("Name %d: ", i + 1);

scanf("%s", names[i]);

}

// Get sort order from user

printf("Enter sort order (A for ascending, D for descending): ");

scanf(" %c", &order);

// Sort the names

n = 5;

for (i = 0; i < n-1; i++) {

for (j = i+1; j < n; j++) {

if ((order == 'A' && strcmp(names[i], names[j]) > 0) || (order == 'D' && strcmp(names[i], names[j]) < 0)) {

// Swap names

strcpy(temp, names[i]);

strcpy(names[i], names[j]);

strcpy(names[j], temp);

}

}

}

// Print the sorted names

printf("\nSorted names:\n");

for (i = 0; i < 5; i++) {

printf("%s\n", names[i]);

}

return 0;

}