LAB SESSION 2:

EXAMPLE 1:

int main() {

int n, i;

int sum = 0;

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

scanf("%d", &n);

printf("Odd natural numbers using for loop: ");

for (i = 1; n > 0; i += 2, n--) {

printf("%d ", i);

sum += i;

}

printf("\nSum of the first %d odd natural numbers: %d\n", n, sum);

return 0;

}

int main() {

int n, i;

int sum = 0;

// Input the value of n from the user

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

scanf("%d", &n);

printf("Odd natural numbers using do-while loop: ");

i = 1;

do {

printf("%d ", i);

sum += i;

i += 2;

n--;

} while (n > 0);

printf("\nSum of the first %d odd natural numbers: %d\n", n, sum);

return 0;

}

int main() {

int n, i;

int sum = 0;

// Input the value of n from the user

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

scanf("%d", &n);

printf("Odd natural numbers using do-while loop: ");

i = 1;

do {

printf("%d ", i);

sum += i;

i += 2;

n--;

} while (n > 0);

printf("\nSum of the first %d odd natural numbers: %d\n", n, sum);

return 0;

}

OUTPUT:



EXAMPLE 2:

int main() {

int rows, spaces;

printf("Enter the number of rows for the pyramid: ");

scanf("%d", &rows);

for (int i = 1; i <= rows; i++) {

for (spaces = 1; spaces <= rows - i; spaces++) {

printf(" ");

}

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

printf("\* ");

}

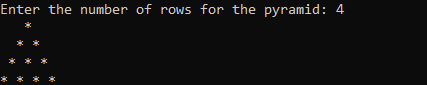
printf("\n");

}

return 0;

}

OUTPUT:



EXAMPLE 3:

int compareStrings(const char \*str1, const char \*str2) {

while (\*str1 != '\0' && \*str2 != '\0') {

if (\*str1 != \*str2) {

return 0;

}

str1++;

str2++;

}

if (\*str1 == '\0' && \*str2 == '\0') {

return 1;

} else {

return 0;

}

}

int main() {

char str1[100], str2[100];

printf("Enter the first string: ");

scanf("%s", str1);

printf("Enter the second string: ");

scanf("%s", str2);

if (compareStrings(str1, str2)) {

printf("The strings are equal.\n");

} else {

printf("The strings are not equal.\n");

}

return 0;

}

OUTPUT:



EXAMPLE 4;

int main() {

char sentence[1000];

printf("Enter a sentence: ");

fgets(sentence, sizeof(sentence), stdin);

printf("Original Sentence: %s", sentence);

for (int i = 0; sentence[i] != '\0'; i++) {

if (islower(sentence[i])) {

sentence[i] = toupper(sentence[i]);

} else if (isupper(sentence[i])) {

sentence[i] = tolower(sentence[i]);

}

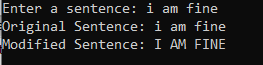
}

printf("Modified Sentence: %s", sentence);

return 0;

}

OUTPUT:



EXAMPLE 5:

int main() {

int array[100];

int n;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

if (n <= 0 || n > 100) {

printf("Invalid array size.\n");

return 1;

}

printf("Enter the elements of the array: ");

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

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

}

printf("Unique elements in the array: ");

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

int isUnique = 1;

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

if (array[i] == array[j]) {

isUnique = 0;

break;

}

}

if (isUnique) {

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

}

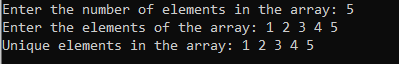
}

printf("\n");

return 0;

}

OUTPUT:



EXAMPLE 6:

struct Distance {

int feet;

float inches;

};

struct Distance addDistances(struct Distance d1, struct Distance d2) {

struct Distance result;

result.feet = d1.feet + d2.feet;

result.inches = d1.inches + d2.inches;

if (result.inches >= 12.0) {

result.feet += 1;

result.inches -= 12.0;

}

return result;

}

int main() {

struct Distance distance1, distance2, sum;

printf("Enter the first distance:\n");

printf("Feet: ");

scanf("%d", &distance1.feet);

printf("Inches: ");

scanf("%f", &distance1.inches);

printf("Enter the second distance:\n");

printf("Feet: ");

scanf("%d", &distance2.feet);

printf("Inches: ");

scanf("%f", &distance2.inches);

sum = addDistances(distance1, distance2);

printf("Sum of distances: %d feet %.2f inches\n", sum.feet, sum.inches);

return 0;

}

OUTPUT:

