Балтийский государственный технический университет «ВОЕНМЕХ»

им. Д. Ф. Устинова

Кафедра О7

«Информационные системы и программная инженерия»

Практическое задание № 2

по дисциплине «Информационные технологии и программирование»

на тему «Программирование ветвлений»

Выполнил:

Студент Карпинская Полина Анатольевна

Группа Е721б

Преподаватель: Удовиченко Андрей Сергеевич

Санкт-Петербург2023

Цель работы – познакомиться с функциями из математической библиотеки, освоить операции отношения, логические операции и условную операцию, изучить работу условного оператора.

Скрипт practical\_task\_2\_1.c:

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

int main() {

float p;

float Q;

printf("Enter the value of the variable p = ");

if (scanf("%f", &p) == 1 && getchar() == '\n') {

Q = (pow(p, 2) - log(abs(pow(p, 1/4) + exp(-p))))/(1 + sin(p)\*cos(p));

printf("Function value Q = %f", Q);

} else printf("Error! Incorrect values entered!");

return 0;

}

Скрипт practical\_task\_2\_2.c:

#include <stdio.h>

#include <stdlib.h>

int main() {

float x, y, f;

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

if (scanf("%f %f", &x, &y) == 2 && getchar() == '\n') {

f = (abs(x - y) > 2) ? (x + 3\*y) : (x == y) ? 3 : 5\*x\*y;

printf("f(x,y) = %.2f", f);

} else printf("Error! Incorrect values entered!");

return 0;

}

Скрипт practical\_task\_2\_3.c:

#include <stdio.h>

int main() {

float x1, y1, x2, y2, x3, y3;

printf("Enter the coordinates of the first point (x1, y1)\n");

printf("Enter the coordinates of the second point (x2, y2)\n");

printf("Enter the coordinates of the third point (x3, y3)\n");

if (scanf("%f%f%f%f%f%f", &x1, &y1, &x2, &y2, &x3, &y3) == 6 && getchar() == '\n') {

float side1 = (x1 - x2) \* (x1 - x2) + (y1 - y2) \* (y1 - y2);

float side2 = (x2 - x3) \* (x2 - x3) + (y2 - y3) \* (y2 - y3);

float side3 = (x3 - x1) \* (x3 - x1) + (y3 - y1) \* (y3 - y1);

if (side1 + side2 > side3 && side2 + side3 > side1 && side3 + side1 > side2) {

printf("Yes, these points can be the vertices of a triangle");

} else printf("No, these points can't be the vertices of a triangle");

} else printf("Error! Incorrect values entered!");

return 0;

}

Ответы на контрольные вопросы:

1. Условная операция и условный оператор имеют схожую концепцию, но некоторые отличия. Условная операция представляет собой оператор, который возвращает одно из двух значений в зависимости от значения условия. Условный оператор, с другой стороны, позволяет выполнить набор инструкций только в том случае, если условие истинно, иначе выполнить другой набор инструкций.

2. Полная форма условного оператора выглядит следующим образом:

if (условие) {

инструкции, выполняемые при истинном условии

} else {

инструкции, выполняемые при ложном условии

}

В неполной форме условного оператора отсутствует блок инструкций для ложного условия:

if (условие) {

инструкции, выполняемые при истинном условии

}

3. Неполная форма условной операции не существует. Условная операция всегда должна иметь два возможных результата - один для истинного условия и один для ложного условия.

4. Если при выполнении условия выполняется оператор return, то блок else можно опустить, поскольку выполнение программы завершается после выполнения return.

5. Выражения, определяющие условия в условном операторе или условной операции, должны иметь логический тип. В C типы данных, такие как int или bool, могут использоваться в качестве условий, где 0 или false соответствуют ложному условию, а все остальные значения соответствуют истинному условию.

6. В условной конструкции истинным считается любое значение, отличное от нуля или false. Ложными считаются 0 или false.

7. Операции отношения включают следующие операторы: == (равно), != (не равно), < (меньше), > (больше), <= (меньше или равно) и >= (больше или равно). Они используются для сравнения значений и возвращают логическое значение истинности или ложности.

8. Операция "==" используется для сравнения двух значений на равенство, тогда как операция "=" используется для присвоения значения одной переменной другой.

9. Логические операции включают операторы && (логическое И), || (логическое ИЛИ) и ! (логическое отрицание). Обычно оператор ! имеет самый высокий приоритет, за ним следуют &&, а затем ||.

10. Операцию "&&" можно заменить операцией "&" (побитовое И), но такая замена будет некорректной, поскольку побитовое И не учитывает логическую истинность значений.

11. Операцию "||" нельзя заменить другой операцией, поскольку логическое ИЛИ имеет свою специфику и представляет комбинированное значение истинности или ложности.

12. Значение выражения отношения или логического выражения может быть логическим значением: true (истина) или false (ложь).

13. Для сравнения вещественных чисел на равенство рекомендуется использовать допустимую погрешность. Например, сравнение двух вещественных чисел a и b может быть выполнено следующим образом: if (fabs(a - b) < epsilon), где epsilon - это небольшое положительное число, определяющее допустимую погрешность.

14. Для проверки вхождения значения в некоторый диапазон можно использовать условие вида if (значение >= минимум && значение <= максимум), где минимум и максимум - это границы диапазона.

15. Чтобы проверить целочисленное значение на равенство нулю, можно использовать условие if (значение == 0).

16. Чтобы проверить отличие целочисленного значения от нуля, можно использовать условие if (значение != 0).

17. Вложение условных операторов применяется, когда необходимо проверить несколько условий последовательно и выполнить определенные инструкции в зависимости от их результатов.

18. Для записи вложенных условных операторов используются вложенные блоки инструкций. Например:

if (условие1) {

инструкции1

if (условие2) {

инструкции2

}

}

19. Оператор выбора позволяет выбрать одну из нескольких альтернативных ветвей выполнения кода в зависимости от значения переменной или выражения. В C оператор выбора реализуется с помощью ключевого слова switch.

20. Оператор выбора с помощью вложенных условных операторов может быть записан следующим образом:

if (значение == 1) {

инструкции1

} else if (значение == 2) {

инструкции2

} else if (значение == 3) {

инструкции3

} else {

инструкции по умолчанию

}