

«Реализация простого конвертера с подмножества Python на Си»

Цели задания:

Реализовать один из вариантов транслирования, конвертер с языка программирования Python на язык программирования Си.

Общая часть

Программа должна читать весь входной текст из стандартного ввода (stdin) с помощью getchar() (getwcahar()) и сохранять его в массив char размерностью 1000.

Программа должна выводить результат в stdout с помощью putchar(). (putwchar())

Нельзя использовать scanf, printf, структуры. И всё остальное, что не пройдено.

Входной код на Python — строго заданного формата (см. Таблицу).

Программа должна получать корректный (рабочий код) на Python, и результат его работы превратить в код на Си.

Использовать операторы перенаправления для получения результата. (Если придумаете как, перенаправленный вывод скомпилировать и запустить).

Таблица конструкций (примеры)

	Конструкция Python	Что должен генерировать конвертер на С
Присваивание	x = 42	int x = 42;
Условный оператор:	if x > 0: y = 1	if (x > 0) y = 1;
Цикл while	while i < 10: i = i + 1	while (i < 10) i = i + 1;
Цикл for	for i in range(10): print(i)	for(size_t i=0;i<10;i++) printf("%lu" , i);
Вывод	print("hello")	printf("%s" , "hello");
Ввод	x = int(input())	scanf("%d", &x);

Критерии оценки

Уровень «Удовлетворительно»

Программа читает входной текст посимвольно.

Корректно распознаёт минимум конструкции (например, `x = int(input())` `print(x)`).

Генерирует эквивалент на языке С (например, `int x = 5;`).

Работает для простых имён переменных (`a`, `x`, `counter`) и целых чисел без знака.

Не использует запрещённые конструкции.

Уровень «Хорошо»

Выполнено всё для «Удовлетворительно»

Поддержка знаковых чисел: $x = -5 \rightarrow \text{int } x = -5;$

Поддержка простых выражений в правой части для оператора присваивания:

$x = a + 1 \rightarrow x = a + 1;$ (без `int`, т.к. переменная уже объявлена)

Для условного оператора и цикла — поддержка простых условий: `if x >= y: ...`
 $\rightarrow \text{if } (x \geq y) \dots;$

Для `print` — поддержка строк в двойных кавычках (без escape-последовательностей).

Уровень «Отлично»

Выполнено всё для «Хорошо»

Программа автоматически определяет, нужно ли объявлять переменную:

Если встретила $x = \dots$ в глобальном контексте — генерирует `int x = ...;`

Если встретила $x = \dots$ после `input()` или в теле условия/цикла — генерирует `x = ...;` (без `int`).

Поддержка многострочного ввода (например, `if ...: на одной строке, тело — на следующей с отступом`).

Корректная обработка пробелов и отступов (игнорирование, но распознавание структуры).

Для `input()` — генерация объявления переменной, если её раньше не было: `age = int(input()) → int age; scanf("%d", &age);`

Требования к оформлению отчёта (формализовано)

Отчёт оформляется в электронной форме и должен содержать следующие разделы:

1. Титульный лист

- По шаблону СибГУТИ
- Название работы: «Генерация и анализ ассемблерного кода программы
- на Си под разные архитектуры» вариант.
- ФИО студента, группа
- Дата сдачи

2. Исходный код конвертера на Си с комментариями.

3. Описание алгоритма: как программа распознаёт конструкцию

4. Примеры работы:

Вход → Выход (1 хороший пример, 1 плохой пример, с объяснением, почему не будет обрабатывать).

5. Вывод: что вы узнали о различиях между Python и Си на уровне синтаксиса и модели выполнения?