Team 3
CS19B001 ADITYA SHARMA CS19B003 AMIT KESARI CS19B013 G Vaibhav

Language Manual: UwU Compiler

Datatypes:

- Using "**let**" keyword for all Integer, Float and String. All will be used with the same datatype specifier keyword.
- "**const**" keyword will be used for contant type variables
- Must be initialized before/if not initialized defaulted to int datatype
- Datatype can be changed (if initially it is int, and the user re-assigns it with string then we will change the type to string)
- Allowed declarations one at a line
  - let num = 10; $* float *$
  - let num = "10"; $* string *$
  - const num = 10; $* constant *$
- Global scope of variables

Operators:

- Arithmetic (**+, -, *, /, %**)
- Boolean (**||, &&, ^, |, &**)
- String (**+**)
- Conditional (**<, >, >=, <=, ==**) (By Values)
- Values parameter of Variable
- Appropriate Errors for Strings

Conditional Statements:

- "**if**" statements
- "**if**", "**if else**", "**else**" statements
- Nested "**if else**"
- Syntax similar to C-based language
- Condition should be a non null non zero value

Loops:

- **"loop"** keyword will be used having the condition first while the looping parameter at the end
- Code Snippet for syntax:
  - i=0;

```
loop(i < 10) {
        print("Hello looper boss")
}(i++)
```

Control Flow:

- "**stop**" will be used to stop a loop execution
- "**continue**" will be used to stop the particular iteration of the loop
- "**return**" will be used to return from a function call

Functions:

- Initialized using "**function**" keyword
- Each function will return something
- Can take at-max 4 number of arguments
- Passed by value
- For functions in which user is not returning anything, 0 will be returned
- **main()** function will start the code

Comments and Delimiters:

- "**;**" will be used in end of each line
- "**$*** -- multi line comments -- ***$**" for multi line comments
- **\t, \n** will be ignored

I/O Operations:

- **print**(a) will be used to print a variable named "a"
- **print**("string") will be used to print a given string
- **print**(a + "string") will be used to concatinate a and string and print
- **input**(a) will be used to take "a" variable as input (by default int)
- **input**(a,1) will be used to input "a" variable as string

Macros:

- Header files inclusion:
    - **#add <filename>**; $* import statement *$

- Macros:
    - **#define VARNAME1::VARNAME2**; $* change varname1 to varname2 *$
        - Example: #define ll :: long long;

Reserved Words:

- let
- const
- if
- loop

- Stop
- continue
- function
- return
- print
- main
- input
- add
- define

Input/Output :

- For input in variable use input statement().
    - Example: input(a);
- For printing a statement use print() function.
    - Example: print("Integer Input:" + i + "\n");

Examples:

1.
```
function main () {
        let a = 10;
        let b = 20;
        let c = 0;

        c = a + b;

        print(c + "\n");

        return 0;
}
```
Output: 30

2.
```
function main() {
        let i = 5;
        input(i);
        print("Integer Input:" + i + "\n");

        input(i,1);
        print("String Input:" + i + "\n");

        return 0;
}
```

Output:
5
Integer Input: 5
Amit
String Input: Amit

3.

```
function main() {
    $* conditions *$
  let k = 0;
  input(k);

    if (k == 0 || k < 2){
            k ++;
  }else{
            k = k + 10;
  }

  print(k);
  return 0;
}
```

Output:
5
15

4.

```
function day_print(){
  let day = 0;
  input(day);
  if(day == 1){
            print("Monday\n");
  }else if(day == 2){
            print("Tuesday\n");
  }else if(day == 3){
            print("Wednesday\n");
  }else if(day == 4){
            print("Thursday\n");
  }else if(day == 5){
            print("Friday\n");
  }else if(day == 6){
            print("Saturday\n");
  }else if(day == 7){
            print("Sunday\n");
  }else{
            print("Invalid Input\n");
  }
  return 0;
}

function main() {
    $* call day *$
  let k = 0;
    day_print();
  return 0;
}
```
Output: 5
Friday

5.
```
function greet() {
  let i=0;
  loop(i < 10) {
          print("Hello sir\n");
  }(i++)
  return 0;
}

function main() {
    $* call greetings *$
  let k = 0;
    greet();
  return 0;
}
```
Output:
Hello sir
Hello sir
Hello sir
Hello sir
Hello sir
Hello sir
Hello sir
Hello sir
Hello sir
Hello sir

6.
```
function main(){
  let arr[3];

  arr[0] = 10;
  arr[1] = 11;
  arr[2] = 12;


  let i = 0;


  loop(i<3){
          print(arr[i]);
  }(i++)

  return 0;
}
```

Output:
10 11 12

7.
```
function main(){
  let arr[3][3][2];

  arr[0][2][1] = 10 + 11;

  let c = 0;

  print(arr[0][2][1]);

  c = c + 10;

  arr[0][2][1] = c;

  print(c + "\n");
  print(arr[0][2][1]);

  return 0;

}
```

Output:
21 10
10

8.
```
function main(){
  $*
          Errors to show
  *$
  let arr[10];
  let a  = 1;
  let b = 2;

  let c = a  b;

  print(a]);

  arr[2]] = 0;


  a = ;

}
```

Output: Error on line 18

9.

```
function main(){
  let k = 0;
  let i = 0;

  if(k == 0){
          i=0;
          loop(i<10){
          print(i+"\n");
          }(i++)
  }else{
          i=0;
          loop(i<10){
          print("Bad \n");
          }(i++)
  }

  return 0;
}
```

Output:

```
0
1
2
3
4
5
6
7
8
9
```

10.

```
#add <./examples/pgm_pre.uwu>
#define pp::print
#define l::let

function main(){
  l i = 0;

  say_hi_from_out();

  return 0;
}
```

Output:
hi