PIAIC IOT ASSIGNMENT 4

Queestion 1:

|  |
| --- |
| use std::io::{self,Write};    fn main() {  print!("Enter the number:");    let mut input\_variable = String::new();    let \_ =io::stdout().flush();    io::stdin().read\_line(&mut input\_variable).expect("Failed to read line");    let int\_data: i32 = input\_variable.trim().parse().expect("Please type a number!");  checker\_condition(int\_data);  }    fn checker\_condition(int\_data: i32) {    if (int\_data>0){  print!("Number is positive \n");  }  else if(int\_data<0) {  print!("Number is negative \n ");  }  else{ print!("Number is equal to zero \n");  }  } |

Question 2:

|  |
| --- |
| use std::io::{self,Write};    fn main() {  print!("Enter the Number in Integer Datatype:");    let mut input\_variable = String::new();    let \_ =io::stdout().flush();    io::stdin().read\_line(&mut input\_variable).expect("Failed to read line");    let int\_data: i32 = input\_variable.trim().parse().expect("Please type a number!");    print!("Enter the Number in Float Datatype:");    let mut input\_variable\_1 = String::new();    let \_ =io::stdout().flush();    io::stdin().read\_line(&mut input\_variable\_1).expect("Failed to read line");    let int\_data\_1: f32 = input\_variable\_1.trim().parse().expect("Please type a number!");    print!("Enter the Number in Boolean Datatype (Hint:true / false):");    let mut input\_variable\_2 = String::new();    let \_ =io::stdout().flush();    io::stdin().read\_line(&mut input\_variable\_2).expect("Failed to read line");    let int\_data\_2: bool = input\_variable\_2.trim().parse().expect("Please type a number!");    datatype\_function(int\_data,int\_data\_1,int\_data\_2);  }    fn datatype\_function(int\_data: i32, int\_data\_1: f32,int\_data\_2:bool) {  println!("The value of Integer\_Number is: {}", int\_data);  println!("The value of Float\_Number is: {}", int\_data\_1);  println!("The value of Boolean\_Number is: {}", int\_data\_2);  } |

Question 3:

|  |
| --- |
| use std::io::{self,Write};    fn main() {  print!("Enter the Unsigned Number:");    let mut input\_variable = String::new();    let \_ =io::stdout().flush();    io::stdin().read\_line(&mut input\_variable).expect("Failed to read line");    let int\_data: u32 = input\_variable.trim().parse().expect("Please type a number!");  let variable = print(int\_data);  print!("{:?}",variable);  }  fn print(x:u32) -> (u32,u32){  println!("{}",x);  (x,x\*x)  } |