Task_1.sml

```
(* ----------------------QUESTION-1--------------------- *)
fun factor(x,y): bool =
    if y mod x = 0 then true
    else false

(* ----------------QUESTION-2 ---------------------*)
fun is_prime(0) = false
  | is_prime(1) = false
  | is_prime(n) =
      let
          fun check_divisor(d) =
              if d * d > n then true
              else if n mod d = 0 then false
              else check_divisor(d + 1)
      in
          check_divisor(2)
      end


(* -----------------QUESTION-3 ---------------------*)
fun gcd(0, y) = y
      | gcd(x, 0) = x
      | gcd(x, y) = gcd(y, x mod y)


(* -----------------QUESTION-4 ---------------------*)
fun perfect(x)=
    let
        fun sum_of_divisors(n,d,res)=if d>=n then res
                                     else if n mod d=0 then sum_of_divisors(n,d+1,res+d)
                                     else sum_of_divisors(n,d+1,res)
    in sum_of_divisors(x,1,0)=x
    end;


(* -----------------QUESTION-5 ---------------------*)
fun sum_of_divisors(n) =
    let
        fun ans(d, res) =
            if d >= n then res
            else if n mod d=0 then ans(d+1,res+d)
            else ans(d+1,res)
    in
        ans(1,0)
    end;

fun amicable(x, y) =
    let
```

```
        val sum_x = sum_of_divisors(x)
        val sum_y = sum_of_divisors(y)
    in
        sum_x = y andalso sum_y=x
    end;

(* ------------------QUESTION-6 ----------------------*)
fun occur([],_)=0
    | occur(h::t,x)=(if h=x then 1 else 0)+occur(t,x);

(* ------------------QUESTION-7 ----------------------*)
fun primes(nums)=
    let
        fun prime(n)=is_prime(n)
        fun res([],p_list)=p_list|
                        res(h::t,p_list)=if prime(h) then res(t,h::p_list)
                                     else res(t,p_list)
    in
        res(nums,[])
    end;

(* ------------------QUESTION-8 ----------------------*)
fun prime_factors(n)=
    let
        fun helper(n,d,res)=if n<2 then res
                            else if n mod d = 0 then helper(n div d,d,d::res)
                            else helper(n,d+1,res)
    in
        helper(n,2,[])
    end

(* ------------------QUESTION-9 ----------------------*)
fun merge([],ys)=ys
    | merge(xs,[])=xs
    | merge(x::xs , y::ys)=if x<y then x::merge(xs,y::ys)
                           else y::merge(x::xs,ys);
val merged1=merge([1,4,5,12],[2,6,7]);

(* ------------------QUESTION-10----------------------*)
fun reverse([])=[]
    | reverse(x::xs)=reverse(xs)@[x];
val reverse1=reverse([1,4,5]);


(* ------------------QUESTION-11----------------------*)
fun pi(a,b,f)=
    if a>b then 1
    else f(a)*pi(a+1,b,f);
val result=pi(2,4,fn x=>x);

```

```
(* ------------------QUESTION-12----------------------*)
fun digits 0=[]
  | digits n=digits (n div 10) @ [n mod 10];

(* ------------------QUESTION-13----------------------*)
fun sumdigits 0 =0
    |sumdigits n=(n mod 10)+sumdigits(n div 10);
fun digitalroot n=
    if n<10 then n
    else digitalroot(sumdigits n);
fun additivePersistence n=
    let
    fun helper(n,count)=
        if n<10 then count
        else helper(sumdigits n,count + 1)
    in
        helper(n,0)
    end;

```

TASK_1 OUTPUT SCREENSHOTS

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val factor = fn : int * int -> bool
- factor(12,4);
val it = false : bool
- factor(4,12);
val it = true : bool
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val is_prime = fn : int -> bool
- is_prime(12);
val it = false : bool
- is_prime(3);
val it = true : bool
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val sum_of_divisors = fn : int -> int
val amicable = fn : int * int -> bool
- amicable(220,284);
val it = true : bool
- amicable(23,45);
val it = false : bool
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val gcd = fn : int * int -> int
- gcd(12,3);
val it = 3 : int
- gcd(12,2);
val it = 2 : int
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val perfect = fn : int -> bool
- perfect(6);
val it = true : bool
- perfect(7);
val it = false : bool
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
Task_1.sml:57.26 Warning: calling polyEqual
val occur = fn : ''a list * ''a -> int
- occur([1,2,2,3,4,2],2);
val it = 3 : int
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val is_prime = fn : int -> bool
val primes = fn : int list -> int list
- primes([1,2,3,4,5,6,7]);
val it = [7,5,3,2] : int list
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val prime_factors = fn : int -> int list
- prime_factors(26);
val it = [13,2] : int list
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val merge = fn : int list * int list -> int list
val merged1 = [1,2,4,5,6,7,12] : int list
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val reverse = fn : 'a list -> 'a list
val reverse1 = [5,4,1] : int list
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val pi = fn : int * int * (int -> int) -> int
val result = 24 : int
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val digits = fn : int -> int list
- digits 123;
val it = [1,2,3] : int list
-
```

```
PS D:\TDP LAB\SML> cd "d:\TDP LAB\SML\" ; if ($?) { sml Task_1.sml }
Standard ML of New Jersey (32-bit) v110.96 [built: Fri Dec 13 15:22:22 2019]
[opening Task_1.sml]
val sumdigits = fn : int -> int
val digitalroot = fn : int -> int
val additivePersistence = fn : int -> int
- digitalroot 9876;
val it = 3 : int
- additivePersistence 9876;
val it = 2 : int
```