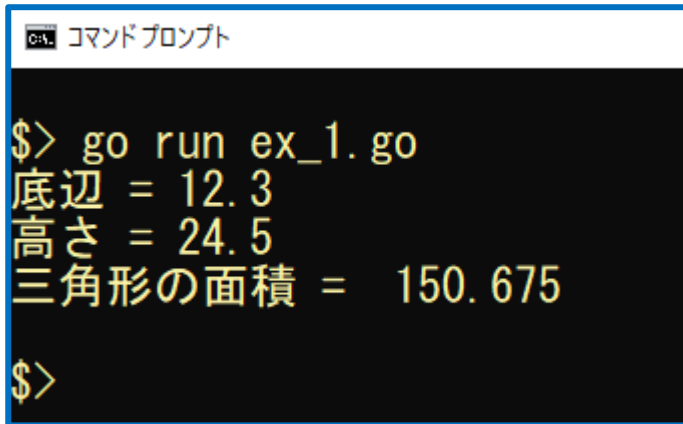


問題1

実行結果



```
cmd コマンドプロンプト

$> go run ex_1.go
底辺 = 12.3
高さ = 24.5
三角形の面積 = 150.675

$>
```

未完成コード ファイル名:ex_1.go

```
/**
 * 三角形の面積を求めて表示する
 */
package main

import "fmt"

func main() {
    // 底辺(bottom)、高さ(height)を宣言、型 float64
    _____(1)_____
    // 面積(area)を宣言、型 float64
    _____(2)_____

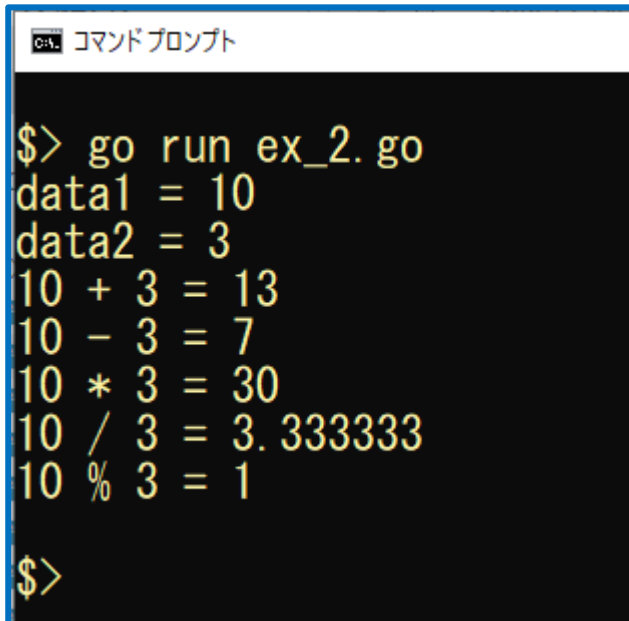
    // キーボードから底辺と高さを入力
    // 入力促進
    _____(3)_____
    // キーボードから底辺を入力し、変数 bottom に保存
    _____(4)_____
    // 入力促進
    _____(5)_____
    // キーボードから高さを入力
    _____(6)_____

    // 三角形の面積を求め area に代入
    _____(7)_____

    // 三角形の面積を表示
    _____(8)_____
}
```

問題2

実行結果



```
コマンドプロンプト

$> go run ex_2.go
data1 = 10
data2 = 3
10 + 3 = 13
10 - 3 = 7
10 * 3 = 30
10 / 3 = 3.333333
10 % 3 = 1

$>
```

未完成コード ファイル名:ex_2.go

```
/**
 * キーボードから二つの整数値を入力し
 * 四則演算を行い結果を表示
 */
package main

import "fmt"

func main() {
    // 変数 data1, data2 を宣言(型 int32)
    (1)_____
    // 変数 addAns, subAns, mulAns, modAns を宣言(型 int32)
    (2)_____
    // 変数 divAns を宣言(型 float32)
    (3)_____

    // 入力促進
    (4)_____
    // キーボードから1つ目の整数値を入力
    (5)_____
    // 入力促進
    (6)_____
    // キーボードから2つ目の整数値を入力
    (7)_____

    // 四則演算
    // 加算
```

```
    (8) _____  
    // 減算  
    (9) _____  
    // 乗算  
    (10) _____  
    // 除算  
    (11) _____  
    // 剰余  
    (12) _____  
  
    // 四則演算結果を表示  
    // 加算  
    (13) _____  
    // 減算  
    (14) _____  
    // 乗算  
    (15) _____  
    // 除算  
    (16) _____  
    // 剰余  
    (17) _____  
}
```

問題3

実行結果

```
cmd コマンドプロンプト

$> go run ex_3.go
身長 (cm単位) を入力 = 168.7
体重 (kg単位) を入力 = 54.4
BMI = 19.114755
適正体重 = 62.611318

$>
```

未完成コード

```
/**
 * キーボードから身長(cm単位)、体重(kg 単位)
 * を入力し BMI 値と適正体重を求めて表示する
 * 但し、BMI 値は以下の計算を行う
 *   BMI = 体重kg / (身長m)2
 *   適正体重 = 22 × (身長m)2
 */
package main

import "fmt"

func main() {
    // 変数 height, weight の宣言(型 float64)
    _____(1)_____

    // 身長(cm 単位)を入力
    // 入力促進
    _____(2)_____
    // 身長(cm単位)を入力し、変数 height に保存
    _____(3)_____

    // 体重(kg単位)を入力
    // 入力促進
    _____(4)_____
    // 体重(kg単位)を入力し、変数 weight に保存
    _____(5)_____

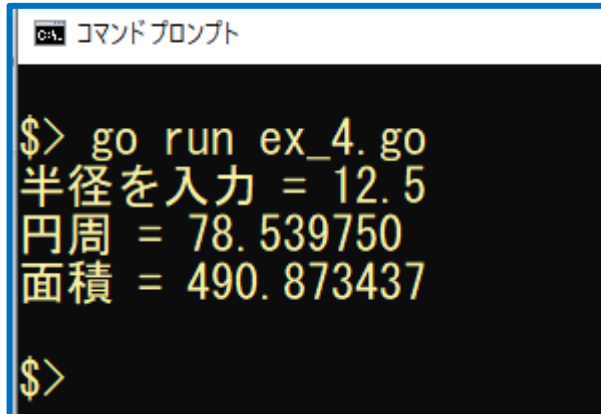
    // 変数 height をm 換算し、変数 heightM に代入
    _____(6)_____

    // BMI 値を計算し、変数 bmi に代入
    _____(7)_____
    // 適正体重を計算し、変数 suitableWeight に代入
```

```
_____(8)_____  
// BMI 値を表示  
_____(9)_____  
// 適正体重を表示  
_____(10)_____  
}
```

問題4

実行結果



```
C:\> go run ex_4.go
半径を入力 = 12.5
円周 = 78.539750
面積 = 490.873437
$>
```

未完成コード ファイル名:ex_4.go

```
/**
 * キーボードから円の半径を入力し
 * 円周と面積を求めて表示する
 */
package main

import "fmt"

// 円周率 PAI (定数)3.14159 の宣言
_____(1)_____

func main() {
    // 変数 radius (半径)の宣言(型 float64)
    _____(2)_____

    // キーボードから半径を入力
    // 入力促進
    _____(3)_____
    // キーボードから半径を入力し、変数 radius に保存
    _____(4)_____

    // 円周を計算し、変数 circumference に保存
    _____(5)_____
    // 面積を計算し、変数 area に保存
    _____(6)_____

    // 円周を表示
    _____(7)_____
    // 面積を表示
    _____(8)_____
}
```

問題5

実行結果

```
cmd コマンドプロンプト

$> go run ex_5.go
整数値を入力 = 13
13 は奇数

$> go run ex_5.go
整数値を入力 = 14
14 は偶数

$>
```

未完成コード ファイル名:ex_5.go

```
/**
 * キーボードから整数値を入力し
 * 偶数か奇数かを判断する
 */
package main

import "fmt"

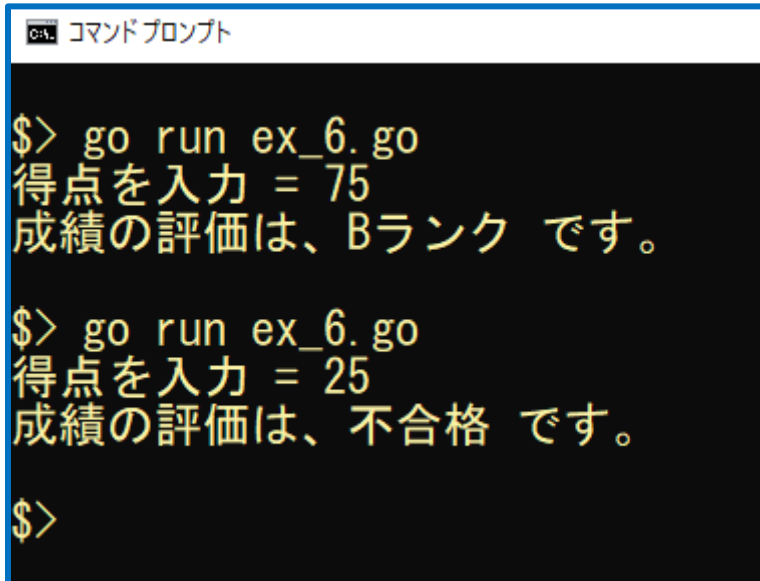
func main() {
    // 変数 data の宣言(型 int32)
    _____(1)_____

    // 入力促進
    _____(2)_____
    // キーボードから整数値を入力
    _____(3)_____

    // 整数値が偶数か奇数かを判断
    if _____(4)_____ {
        // 偶数だったら
        _____(5)_____
    } _____(6)_____ {
        // 奇数だったら
        _____(1)_____
    }
}
```

問題6

実行結果



```
Ca. コマンドプロンプト

$> go run ex_6.go
得点を入力 = 75
成績の評価は、Bランク です。

$> go run ex_6.go
得点を入力 = 25
成績の評価は、不合格 です。

$>
```

未完成コード ファイル名:ex_6.go

```
/**
    キーボードから得点を入力し
    成績の評価を表示する
    但し、得点の評価は以下のとおりとする
        90 点以上 S ランク
        80 点以上 A ランク
        70 点以上 B ランク
        50 点以上 C ランク
        40 点以上 D ランク
        40 点未満 不合格
*/
package main

import "fmt"

func main() {
    // 変数 score (得点)を宣言(型 int32)
    _____(1)_____
    // 変数 evaluation (評価の文字列)を宣言(型 string)
    _____(2)_____

    // キーボードから得点を入力
    // 入力促進
    _____(3)_____
    // キーボードから得点を入力し変数 score に保存
    _____(4)_____

    // 得点の評価
    _____(5)_____ {
```



```
// 90 点以上するとき
evaluation = "S ランク"
} _____(6)_____ {
// 80 点以上するとき
evaluation = "A ランク"
} _____(7)_____ {
// 70 点以上するとき
evaluation = "B ランク"
} _____(8)_____ {
// 50 点以上するとき
evaluation = "C ランク"
} _____(9)_____ {
// 40 点以上するとき
evaluation = "D ランク"
} _____(10)_____ {
// 40 点未満の時
evaluation = "不合格"
}

// 成績評価を表示
_____ (11) _____
}
```

問題7

実行結果

```
cmd コマンドプロンプト

$> go run ex_7.go
チャンネル番号入力 = 8
チャンネル番号 8 に対応するテレビ局名は、フジテレビ

$> go run ex_7.go
チャンネル番号入力 = 5
チャンネル番号 5 に対応するテレビ局名は、ありません

$>
```

未完成コード ファイル名:ex_7.go

```
/**
 * キーボードからチャンネル番号を入力し
 * 対応するテレビ局名を表示する
 */
package main

import "fmt"

func main() {
    // 変数 channelNumber (チャンネル番号)の宣言(型 unit32)
    (1) _____
    // 変数 tvStationName (テレビ局名)の宣言(型 string)
    (2) _____

    // キーボードからチャンネル番号を入力
    // 入力促進
    (3) _____
    // キーボードからチャンネル番号を入力し、変数 channelNumber に保存
    (4) _____

    // チャンネル番号に対応するテレビ局名を選択
    switch (5) _____ {
        (6) _____:
            tvStationName = "NHK 総合"
        (7) _____:
            tvStationName = "NHK 教育"
        (8) _____:
            tvStationName = "日本テレビ"
        (9) _____:
    }
```

```
    tvStationName = "TBS"  
    _____(10)_____  
    tvStationName = "フジテレビ"  
    _____(11)_____  
    tvStationName = "朝日テレビ"  
    _____(12)_____  
    tvStationName = "東京 12 チャンネル"  
    _____(13)_____  
    tvStationName = "ありません"  
}  
  
// チャンネル番号に対応するテレビ局名を表示  
    _____(14)_____  
}
```

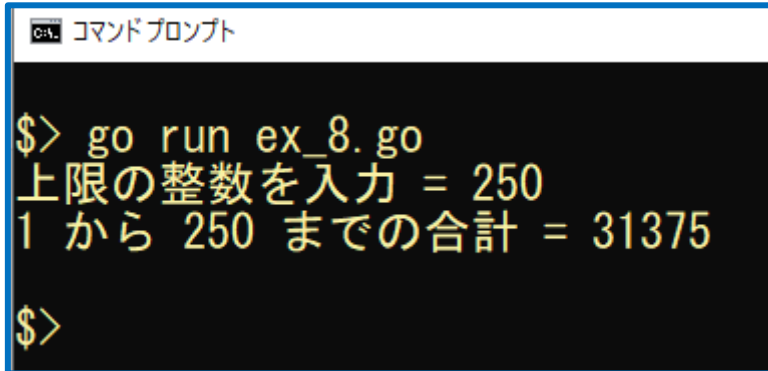
問題7(その2)

問題6のプログラムを switch 文を使用して実装してください。

ファイル名: ex6_2.go

問題8

実行結果



```
C:\> $> go run ex_8.go
上限の整数を入力 = 250
1 から 250 までの合計 = 31375
$>
```

未完成コード ファイル名:ex_8.go

```
/**
 * キーボードから整数を入力し
 * 1からその入力した整数までの整数値の合計を求めて
 * 表示する
 */
package main

import "fmt"

func main() {
    // 変数 limti(上限の整数) を宣言(型 int)
    _____(1)_____
    // 変数 total(合計)を宣言(型 int)と 0 に初期化
    _____(2)_____

    // 上限の整数を入力
    // 入力促進
    _____(3)_____
    // キーボードから整数を入力し、変数 limit に保存
    _____(4)_____

    // 1から上限の整数までの整数値の合計を求める
    _____(5)_____ {
        total += i
    }

    // 合計値を表示
    _____(6)_____
}
```

問題9

実行結果

```
cmd コマンドプロンプト

$> go run ex_9.go
上限の整数を入力 = 250
1 から 250 までの奇数の合計 = 15625

$>
```

未完成コード ファイル名:ex_9.go

```
/**
 * ex 8.go において
 * 1から上限の整数までの奇数の合計を求めて表示する
 */
package main

import "fmt"

func main() {
    // 変数 limti (上限の整数) を宣言(型 int)
    _____(1)_____
    // 変数 total (合計)を宣言(型 int)と 0 に初期化
    _____(2)_____

    // 上限の整数を入力
    // 入力促進
    _____(3)_____
    // キーボードから整数を入力し、変数 limit に保存
    _____(4)_____

    // 1から上限の整数までの整数値の合計を求める
    _____(5)_____ {
        _____(6)_____ {
            total += i
        }
    }

    // 合計値を表示
    _____(7)_____
}
```

問題10

実行結果

```
コマンドプロンプト

$> go run ex_10.go
上限の整数を入力 = 250
倍数を入力 = 7
1 から 250 までの 7 の倍数の合計 = 4410

$>
```

未完成コード ファイル名:ex_10.go

```
/**
ex_9.go において
上限の整数値を入力した後、さらに倍数を表す整数値を
キーボードから入力し、1から上限の整数までの入力された
倍数の合計を求めて表示する
*/
package main

import "fmt"

func main() {
    // 変数 limti(上限の整数) を宣言(型 int)
    _____(1)_____
    // 変数 total(合計)を宣言(型 int)と 0 に初期化
    _____(2)_____
    // 変数 mod (倍数)を宣言(型 int)
    _____(3)_____

    // 上限の整数を入力
    // 入力促進
    _____(4)_____
    // キーボードから整数を入力し、変数 limit に保存
    _____(5)_____

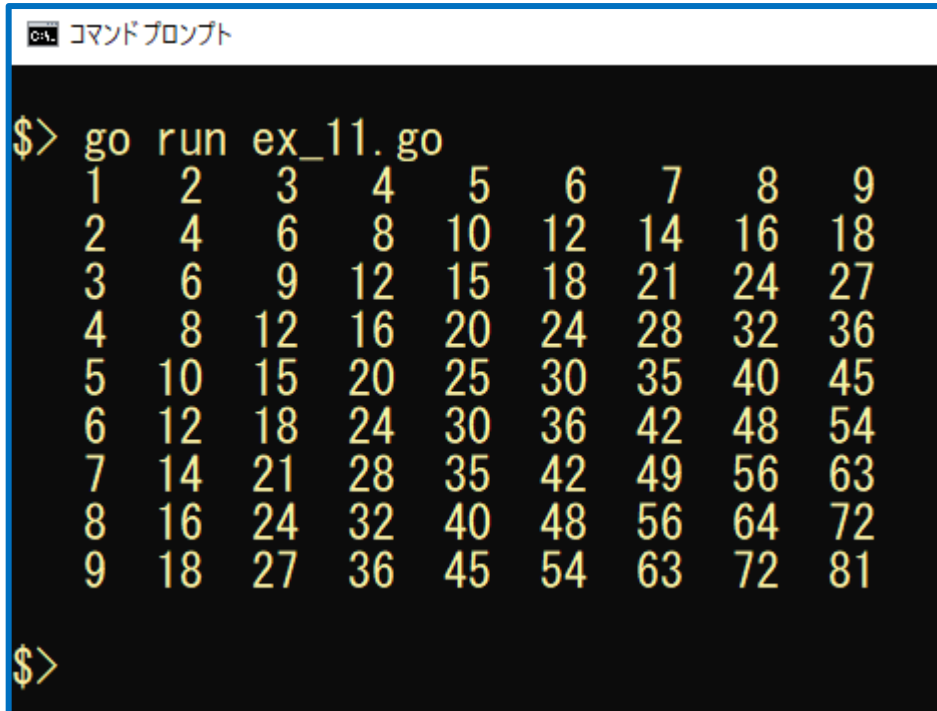
    // 倍数を入力
    // 入力促進
    _____(6)_____
    // キーボードから倍数を入力し、変数 mod に保存
    _____(7)_____

    // 1から上限の整数までの倍数の合計を求める
    _____(8)_____ {
        _____(9)_____ {
            total += i
        }
    }
}
```

```
    }  
  }  
  
  // 合計値を表示  
  _____(10)_____  
}
```

問題11

実行結果



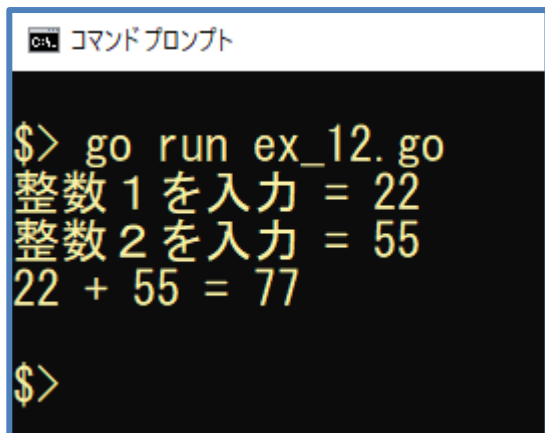
```
$> go run ex_11.go
 1   2   3   4   5   6   7   8   9
 2   4   6   8  10  12  14  16  18
 3   6   9  12  15  18  21  24  27
 4   8  12  16  20  24  28  32  36
 5  10  15  20  25  30  35  40  45
 6  12  18  24  30  36  42  48  54
 7  14  21  28  35  42  49  56  63
 8  16  24  32  40  48  56  64  72
 9  18  27  36  45  54  63  72  81

$>
```

ファイル名: ex_11.go

問題12

実行結果



```
C:\> go run ex_12.go
整数 1 を入力 = 22
整数 2 を入力 = 55
22 + 55 = 77
C:\>
```

未完成コード ファイル名:ex_12.go

```
/**
 * ポインタを使用したたし算
 */
package main

import "fmt"

func main() {
    // 変数 number1, number2 (型 int) を宣言
    (1)
    // ポインタ変数 ptr1, ptr2 (型 *int) を宣言
    (2)

    // 2つの整数値をキーボードから入力
    // 入力促進
    (3)
    // キーボードから整数値を入力し、変数 number1 に保存
    (4)
    // 入力促進
    (5)
    // キーボードから整数値を入力し、変数 number1 に保存
    (6)

    // 変数 number1 のアドレスをポインタ変数 ptr1 に代入
    (7)
    // 変数 number2 のアドレスをポインタ変数 ptr2 に代入
    (8)

    // ポインタ変数 ptr1, ptr2 を使用して、キーボードから入力した
    // 2つの整数値のたし算をして、変数 addAns に代入
    (9)
    // たし算の結果を表示
    (10)
}
```