

情報処理応用 (木村) 2023-04-28

login:

fajifaji

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

1. 数 x, y を引数にとり、それらを足した数を戻り値とする関数 `add2(x, y)`.

2. 整数 x を引数にとり、それが偶数であれば `True`、奇数だったら `False` を返す関数 `is_even(x)`.

3. $n + (n+1) + (n+2) + \dots + m$ を求める関数 `sum_int(n, m)`.

4. リスト `xs` の各要素を n 倍したリストを返す関数 `times_n(xs, n)`.

5. 整数リスト `xs` に含まれる偶数だけを抜き出す `evens_only(xs)`.

→ `def sum_int(n, m):`
`a = []`
`for i in range(n, m+1):`
`a.append(i)`
`return sum(a)`
→ `def evens_only(xs):`
`return xs[::2]`

1 `def add2(x, y):`
`return (x + y)`

2 `def is_even(x):`
`if x % 2 == 0:`
`return True`
`else:`
`return False`

4. `def times_n(xs, n):`

`a = []`
`for i in xs:`
`a.append(i * n)`
`return a`

10

情報処理応用 (木村) 2023-04-28

login: avry2620

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

1. 数 x, y を引数にとり、それらを足した数を戻り値とする関数 $\text{add2}(x, y)$.
2. 整数 x を引数にとり、それが偶数であれば True、奇数だったら False を返す関数 $\text{is_even}(x)$.
3. $n + (n+1) + (n+2) + \dots + m$ を求める関数 $\text{sum_int}(n, m)$.
4. リスト xs の各要素を n 倍したリストを返す関数 $\text{times}_n(xs, n)$.
5. 整数リスト xs に含まれる偶数だけを抜き出す $\text{evens_only}(xs)$.

if $x \% 2 == 0$:
 return True
else:
 return False

1 def add2(x, y):
 return x + y

2 def is_even(x):
 return x % 2 == 0

5 def evens_only(xs):
 rt = (xs % 2 == 0)
 return filter(even_only, rt)

情報処理応用 (木村) 2023-04-28

login: umbreon

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

1. 数 x, y を引数にとり、それらを足した数を戻り値とする関数 $\text{add2}(x, y)$.
2. 整数 x を引数にとり、それが偶数であれば True 、奇数だったら False を返す関数 $\text{is_even}(x)$.
3. $n + (n+1) + (n+2) + \dots + m$ を求める関数 $\text{sum_int}(n, m)$.
4. リスト xs の各要素を n 倍したリストを返す関数 $\text{times_n}(xs, n)$.
5. 整数リスト xs に含まれる偶数だけを抜き出す $\text{evens_only}(xs)$.

1.

```
def add2(x, y):  
    return x + y
```

2.

```
def is_even(x):  
    return x % 2 == 0
```

5.

```
def evens_only(xs):  
    ret = []  
    for i in xs:  
        if i % 2 == 0:  
            ret.append(i)  
    return ret
```


情報処理応用 (木村) 2023-04-28

login: shinjib

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

1. 数 x, y を引数にとり、それらを足した数を戻り値とする関数 $\text{add2}(x, y)$.
2. 整数 x を引数にとり、それが偶数であれば True 、奇数だったら False を返す関数 $\text{is_even}(x)$.
3. $n + (n+1) + (n+2) + \dots + m$ を求める関数 $\text{sum_int}(n, m)$.
4. リスト xs の各要素を n 倍したリストを返す関数 $\text{times_n}(xs, n)$.
5. 整数リスト xs に含まれる偶数だけを抜き出す $\text{evens_only}(xs)$.

```
1. def add2(x, y):  
    return x + y
```

```
2. def is_even(x):  
    if x % 2 == 0:  
        return True  
    return False
```

```
5. def evens_only(xs):  
    a = []  
    for i in xs:  
        if i % 2 == 0:  
            a.append(i)  
    return a
```

~~X~~ == 正解。

△
○

情報処理応用（木村） 2023-04-28

login: Yogurt

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

- ① 数 x, y を引数にとり、それらを足した数を戻り値とする関数 `add2(x, y)`.
- ② 整数 x を引数にとり、それが偶数であれば `True`、奇数だったら `False` を返す関数 `is_even(x)`.
- ③ $n + (n+1) + (n+2) + \dots + m$ を求める関数 `sum_int(n, m)`.
4. リスト `xs` の各要素を n 倍したリストを返す関数 `times_n(xs, n)`.
5. 整数リスト `xs` に含まれる偶数だけを抜き出す `evens_only(xs)`.

```
1. def add2(x, y):  
    return x + y
```

```
2. def is_even(x):  
    if x % 2 == 0:  
        return True  
    else:  
        return False
```

```
3. def sum_int(n, m):  
    return (n+m)*(m-n+1)/2
```

合

情報処理応用 (木村) 2023-04-28

login: havapeko

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

1. 数 x, y を引数にとり、それらを足した数を戻り値とする関数 $\text{add2}(x, y)$.

2. 整数 x を引数にとり、それが偶数であれば True、奇数だったら False を返す関数 $\text{is_even}(x)$.

3. $n + (n+1) + (n+2) + \dots + m$ を求める関数 $\text{sum_int}(n, m)$.

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ & & & & & \uparrow \\ & & & & & (1+5) \times \\ & & & & & 5-2+1 \end{array}$$

4. リスト xs の各要素を n 倍したリストを返す関数 $\text{times}_n(xs, n)$.

5. 整数リスト xs に含まれる偶数だけを抜き出す $\text{evens_only}(xs)$.

```
1 def add2(x, y):  
    return x+y
```

```
2 def is_even(x):  
    # '2で割った余りが0' if True, otherwise False  
    return x % 2 == 0
```

```
3 def sum_int(n, m):  
    # (初項 + 末項) × 項数 × 1/2  
    return (n+m) * (m-n+1) / 2
```

```
4 def times_n(xs, n):  
    return [i*n for i in xs]
```

```
5 def evens_only(xs):  
    # リスト xs (= xs のうち偶数を追加)  
    ys = []  
    for i in xs:  
        if i % 2 == 0:  
            ys.append(i)  
    return ys
```

合

情報処理応用 (木村) 2023-04-28

login: museru53

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

1. 数 x, y を引数にとり、それらを足した数を戻り値とする関数 $\text{add2}(x, y)$.
2. 整数 x を引数にとり、それが偶数であれば True、奇数だったら False を返す関数 $\text{is_even}(x)$.
3. $n + (n+1) + (n+2) + \dots + m$ を求める関数 $\text{sum_int}(n, m)$.
4. リスト xs の各要素を n 倍したリストを返す関数 $\text{times}_n(xs, n)$.
5. 整数リスト xs に含まれる偶数だけを抜き出す $\text{evens_only}(xs)$.

1. $\text{def add2}(x, y):$
 $\text{return } x + y$

2. $\text{def is_even}(x):$
 $\text{return } x \% 2 == 0$

5. $\text{def evens_only}(xs):$
 $\text{ref} = []$
 for i in xs :
 if $i \% 2 == 0$:
 $\text{ref.append}(i)$
 return ref

$\text{def evens_only}(xs):$
 $\text{ref} = []$
 for

$$\frac{(n+m) \cdot (m-n+1)}{2}$$

5

情報処理応用 (木村) 2023-04-28

login: bindume

以下の5問中、3問を選択し、問題番号明記の上、回答せよ。部分点なし。時間は30分。回答した3問中、2問正解で再試験合格。合格者は2023年の情報処理応用の履修届けを出しなさい。授業には来なくてよい。60点で成績報告する。不合格者は2023年以降の情報処理応用を履修すること。

1. 数 x, y を引数にとり、それらを足した数を戻り値とする関数 $\text{add2}(x, y)$.
2. 整数 x を引数にとり、それが偶数であれば True、奇数だったら False を返す関数 $\text{is_even}(x)$.
3. $n + (n+1) + (n+2) + \dots + m$ を求める関数 $\text{sum_int}(n, m)$.
4. リスト xs の各要素を n 倍したリストを返す関数 $\text{times}_n(xs, n)$.
5. 整数リスト xs に含まれる偶数だけを抜き出す $\text{evens_only}(xs)$.

1.

```
def add2(x, y):  
    return x + y
```

2.

```
def is_even(x):  
    return x % 2 == 0
```

3.

```
def sum_int(n, m):  
    sum = 0  
    for i in range(n, m+1):  
        sum += i  
    return sum
```

合