

令和8（2026）年度 2月入学試験問題

商学・経済学研究科 商学・経済学専攻
修士課程

一 般

専門科目
(財務会計論)

受 験 上 の 注 意

1. 試験開始の合図があるまで試験問題を開いてはいけません。
2. 受験者は、すべて試験監督員の指示に従ってください。
3. 試験時間内の退場はできません。
4. 解答用紙のみ提出してください。
5. 試験時間は90分です。

熊本学園大学大学院

令和8（2026）年度 熊本学園大学大学院修士課程 2月入学試験問題
〔商学・経済学研究科 商学・経済学専攻〕

【 専門科目 】

〔財務会計論〕

〔問〕 我が国の会計基準では、資産の性質に応じて評価基準を使い分ける考え方が採用されている。具体的には、生産や販売など企業の本来の活動に用いる事業用資産は、原則として取得原価で評価される一方、余剰資金の運用として保有する金融資産は、市場価格を基礎とする時価で評価される。なぜこのような考え方が採用されているのか、その理由を詳しく説明しなさい。

令和8（2026）年度
2月入学試験問題

商学・経済学研究科 商学・経済学専攻
修士課程

一 般

専門科目
(動学マクロ経済学)

受 験 上 の 注 意

1. 試験開始の合図があるまで試験問題を開いてはいけません。
2. 受験者は、すべて試験監督員の指示に従ってください。
3. 試験時間内の退場はできません。
4. 解答用紙のみ提出してください。
5. 試験時間は90分です。

熊本学園大学大学院

【 専門科目 】

〔動学マクロ経済学〕

2 期間生存する家計は、次の式で与えられる生涯効用 U を最大にするように、現在(第1期)と将来(第2期)の消費を決定する。

$$U(c, d) = u(c) + \frac{1}{1+\rho} u(d) \quad (1)$$

ここで、 c は現在の消費、 d は将来の消費、 $\rho > 0$ は主観割引率(現在と将来で同じ量を消費したとしても、将来の消費から得られる効用を割り引いて現在の効用よりも小さく評価する際の割引率)である。各期の効用関数 u は、 $u'(\cdot) > 0, u''(\cdot) < 0$ を満たしている。

第1期の家計の予算制約は、 y を所得、 s を貯蓄として

$$y = c + s \quad (2)$$

である。これに対して第2期の家計の予算制約は r を利子率として

$$d = (1 + r)s \quad (3)$$

である。このとき、以下の問いに答えよ。

問1:(2)と(3)から貯蓄 s を消去し、生涯予算制約式を $d =$ の形で求めよ。

問2:(1)で与えられる生涯効用関数を、問1で求めた生涯予算制約式を用いて現在の消費 c のみの関数 $V(c)$ として表現せよ。

問3:効用最大化の条件からオイラー方程式を求め、 $1 + r =$ の形で表せ。

問4:主観割引率が利子率と等しい場合の、現在と将来の消費の特徴を簡潔に述べよ。

問5:現在と将来の最適な消費 c^*, d^* を求めよ。

受験番号					

熊本学園大学大学院
商学・経済学研究科（修士課程）

※氏名は記入しないこと。

解答用紙（動学マクロ経済学）

※解答欄が不足する場合は裏面に記入すること。
裏面に解答する際は、どの解答か分かるよう明記すること。

評点	
	/ 100

計算用紙

令和 8 (2026) 年度
2 月入学試験問題

商学・経済学研究科 商学・経済学専攻
修士課程

一 般

専門科目
(情報処理論)

受 験 上 の 注 意

1. 試験開始の合図があるまで試験問題を開いてはいけません。
2. 受験者は、すべて試験監督員の指示に従ってください。
3. 試験時間内の退場はできません。
4. 解答用紙のみ提出してください。
5. 試験時間は 90 分です。

熊本学園大学大学院

【 専門科目 】

〔情報処理論〕

設問 1 次のプログラムの (1) に入れる正しい答えを、解答群の中から選べ(配点 25 点)

図 1 に示すグラフの頂点には 1 から順に整数で番号が付けられている。グラフは無向グラフであり、各頂点間には高々一つの辺がある。一つの辺は両端の頂点の番号を要素にもつ要素数 2 の整数型の配列で表現できる。例えば、[1,3] は頂点 1 と頂点 3 を端点とする辺を表す。グラフ全体は、グラフに含まれる辺を表す要素数 2 の配列をすべて格納した配列 (以下、辺の配列という) で表現できる。辺の配列の要素数はグラフの辺の個数と等しい。図 1 のグラフは整数型配列の配列[[1,3],[1,4],[3,4],[2,4],[4,5]]と表現できる。

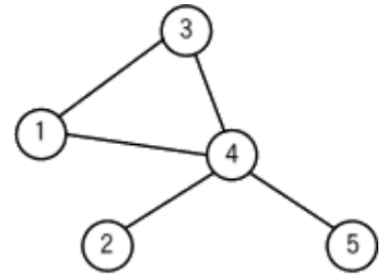


図 1 グラフ例

関数 edgeToMatrix は、辺の配列を隣接配列に変換する。隣接配列とは、グラフに含まれる頂点の個数と等しい行列及び列数の正方行列で、i 行 j 列の成分は頂点 i と頂点 j を結ぶ辺があるとき 1 となり、それ以外は 0 となる。行列の体格成分はすべて 0 で、無向グラフの場合は対称行列になる。図 1 のグラフを表現する隣接行列を図 2 に示す。



図 2

図 1 のグラフを表現する隣接行列

関数 edgeToMatrix は、引数 \$edgeList で辺の配列を、引数 \$nodeNum でグラフの頂点の個数をそれぞれ受け取り、隣接行列を表す整数型の二次元配列を返す。

```
function edgeToMatrix ($edgeList, $nodeNum) {
    $adjMatrix = array();
    for ($j=0; $j< $nodeNum; $j++) {
        $adjMatrix[$j]= array(0,0,0,0,0);
    }
    for ($i=0; $i < count($edgeList); $i++) {
        $u=($edgeList[$i][0]-1);
        $v=($edgeList[$i][1]-1);
        ( 1 )
    }
    return $adjMatrix;
}
```

- (a) \$adjMatrix[\$u][\$u] =1; (b) \$adjMatrix[\$u][\$v] =1; (c) \$adjMatrix[\$v][\$u] =1;
- (d) \$adjMatrix[\$v][\$v] =1;
- (e) \$adjMatrix[\$u][\$u] =1; (f) \$adjMatrix[\$u][\$v] =1;
- \$adjMatrix[\$v][\$v] =1; \$adjMatrix[\$v][\$u] =1;

設問2 次の記述中の (2) に入れる正しい答えを、解答群の中から選べ(配点 25 点)

次の手続き sorttest は、整数型の配列 \$data の、引数 \$first で与えられた要素番号から引数 \$last で与えられた要素番号までの要素を昇順に整理する。ここで \$first < \$last とする。手続き sorttest を sort(1,5)として呼び出すと、*/** α **/*の行を最初に行を実行したときの出力は” (2) ”となる。

```
function sorttest($first, $last) {
    $data = [2,1,3,5,4];
    $pivot = intval($data[(($first-1) + ($last-1))/2]);
    $i = ($first-1);
    $j = ($last-1);
    while (true) {
        while ($data[$i] < $pivot) {
            $i++;
        }
        while ($data[$j] > $pivot) {
            $j--;
        }
        if ($i >= $j) {
            break;
        }
        $datai = $data[$i];
        $dataj = $data[$j];
        $data[$i] = $dataj;
        $data[$j] = $datai;
        $i++;
        $j++;
    }
    /** α **/
    if ($first < ($i-1)) {
        sorttest($first, ($i-1));
    }
    if (($j+1) < $last) {
        sorttest(($j+1), $last);
    }
}
```

- (a) 1 2 3 4 5 (b) 1 2 3 5 4 (c) 2 1 3 4 5 (d) 2 1 3 5 4

設問3 記述中の (3) に入れる正しい答えを、解答群の中から選べ(配点 25 点)

関数 merge は、昇順に整列された整数型の配列 data1 および data2 を受け取り、これらを併合してできる昇順に整列された整数型の配列を返す。

関数 merge を merge([2,3],[1,4])として呼び出すと、*/** α **/*の行は (3)。

```
function merge ($data1, $data2) {
    $n1 = count($data1);
    $n2=count($data2);
    $work=array();
    $i=0;
    $j=0;
    $k=0;
    while (($i < $n1) && ($j < $n2)) {
        if ($data1[$i]<=$data2[$j]) {
            $work[$k] = $data1[$i];
            $i++;
        } else {
            $work[$k]=$data2[$j];
            $j++;
        }
        $k++;
    }
    while ($i < $n1) {
        $work[$k] = $data1[$i];
        $i++;
        $k++;
    }
    while ($j < $n2) {
        $work[$k]=$data2[$j];
        /** α **/
        $k++;
        $j++;
    }
    return $work;
}
merge([2,3],[1,4]);
```

(a) 実行されない (b) 1回実行される (c) 2回実行される (d) 3回実行される

設問4 次の記述中の (4) に入れる正しい答えを、解答群の中から選べ(配点 25 点)

関数 `add` は、引数で指定された正の整数 `$value` を帯域の整数型の配列 `$hashArray` に格納する。格納できなかった場合は `true` を返し、格納できなかった場合は `false` を返す。ここで、整数 `$value` を `$hashArray` のどの要素に格納すべきかを、関数 `calcHash1` 及び `calcHash2` を利用して決める。

手続き `test` は、関数 `add` を呼び出して、`$hashArray` に正の整数を格納する。手続き `test` の処理が終了した直後の `$hashArray` の内容は、(4) である。

```
$hashArray=array();
function add ($value) {
    $i = calcHash1($value);
    global $hashArray;
    if ($hashArray[$i]==-1) {
        $hashArray[$i] = $value;
        return true;
    } else {
        $i = calcHash2($value);
        if ($hashArray[$i]==-1) {
            $hashArray[$i] = $value;
            return true;
        }
    }
}
function calcHash1 ($value) {
    global $hashArray;
    return $value%count($hashArray);
}
function calcHash2 ($value) {
    global $hashArray;
    return ($value+3)%count($hashArray);
}
function test () {
    global $hashArray;
    $hashArray = [-1,-1,-1,-1,-1];
    add(3);
    add(18);
    add(11);
}
(a) [-1,3,-1,18,11]    (b) [-1,11,-1,3,-1]
(c) [-1,18,-1,3,11]   (d) [-1,11,-1,18,-1] (e) [-1,18,11,3,-1]
```

受験番号					

※氏名は記入しないこと。

熊本学園大学大学院
商学・経済学研究科（修士課程）

解答用紙（情報処理論）

[設問 1]

[設問 2]

[設問 3]

[設問 4]

評点	/ 100
----	-------

計算用紙