

午後試験

問 1

装置の MTTF (故障までの平均時間) の違いがシステムの維持費用に与える影響の分析について出題した。四つの解答箇所 a~d のうち, a と b の正答率は高く, 故障台数の予測はおおむね理解されているようであった。c と d は多少複雑な計算が必要なためか, 正答率は平均的であった。

MTTF を計算する際に, 年間故障台数を, 延べ稼働時間 [$1(\text{年/台}) \times 1,200(\text{台}) = 1,200(\text{年})$] 当たりの故障台数ととらえることができれば容易に解答できた。

a と b は, 上述のようなとらえ方をして計算すれば, 正答を導き出すのは容易であった。

c と d は, 総費用の多少を比較すればよいので, 差額を求める式を利用して計算の手間を減らすことができた。

問 2

関係データベースの設計及び SQL 文による操作能力について出題した。全体として正答率は平均的であり, おおむね理解されているようであったが, 正規化の手法を題材とした設問 1 は正答率が低かった。

設問 1 は, アと誤答した受験者が多かった。正規化の効果は理解していても, 第 1 正規化の具体的な操作までは理解できていなかったようである。第 1~3 正規化については, 具体的な操作まで理解してほしい。

設問 2 は, 平均的な正答率であり, 正規化の効果についてほぼ理解できていたようである。

設問 3 及び設問 4 はいずれも正答率が高かった。SELECT 文の抽出条件については, おおむね理解できていたようである。

問 3

ガウスの消去法を理解し, それに基づいて作成されたプログラムを完成することについて出題した。a~c のすべてにおいて, 平均的な正答率であった。

a の誤答としてはイが多かった。〔プログラムの説明〕(1) の説明を理解すれば, ピボットとして k 行 k 列の要素を選択しなければならないことが分かり, ウと正答できた。

b では, 外側の繰返しの添字が i であり, $n, n-1, \dots, 1$ の順に解を求めており, 更に添字の値が n の場合の処理は前の行で終わっていることを理解していれば, イと正答することができた。

c では, 外側の繰返しの添字が i であることから繰返しの添字が j であることが分かり, この繰返しが後退代入の式の中で の部分に対応することを理解できれば, 正答することができた。この では, j を $i+1$ から n まで変化させているので, オが正答となる。

問 4

タブ文字を展開する文字列処理について出題した。全体としての正答率は平均的であった。

設問 1 では, a の正答率が低かった。これは, a の行とその次の行とを一緒に考慮しなければならなかったためであると思われる。この 2 行がプログラムの説明文の(3)の式に相当することを理解し, 図 1 の実行例を参考にすれば正答することができた。ア又はウと誤答した受験者も多かったが, 実際にタブの境界値を当てはめて検証することが重要である。

設問 2 では, d の正答率が低かった。d では, イ又はウと誤答した受験者が多かった。イ及びウでは, タブ位置が出力の文字位置 $Didx$ の値によって変わってしまい, プログラム 2 の意図と異なる。これは具体的な値を用いて実行してみれば, すぐに分かる。

問5

与えられた処理の条件と要求仕様を理解し、正しく実装する能力について出題した。設問によって正答率のばらつきはあるが、全体では高い正答率であった。

設問1では、三つの解答箇所 a~c ともよく理解されていたようであるが、a では製造日ごとに集計することを見落としていたためと思われる誤答イ~エが見受けられた。

設問2では、プログラムの説明に明示されている切捨て誤差の補正について、どの計算で切捨て誤差が発生するかを見落としていたためか、エやオと誤答した受験者が多かった。

設問3は、正答率が低く、ウと誤答した受験者が多かった。〔プログラムの説明〕で明示したように牧場区分“J”、“K”、“L”によって、 w_j 、 w_k 、 w_l のどれかに必要量を設定することになる（ほかの二つは0にする）。また、“集計”では、もはや牧場区分は必要としない。

実際のプログラムの要求仕様は、文書や図表などで与えられる。これを設計に正しく反映するためには、与えられたものをよく読み、ポイントとなる要件を抽出することが重要である。

問6

数当てゲームのプログラムを完成することについて出題した。

a, b は、いずれも制御式で指定する関数及び返却値を選択する設問であった。b では、イやエと誤答した受験者があったが、平均的な正答率であった。これらの設問では、do-while 文の文法を理解していること及びこの関数の仕様を読み取るためにプログラムを追跡して理解することが必要であった。

c~e はゲームのアルゴリズムに関する設問であった。いずれも非常に高い正答率であり、よく理解されていた。

問7

索引ファイルに対する更新のうち、データの追加・削除処理について出題した。

a では、アと誤答した受験者が多かった。b ではア、c ではウの誤答が多かったことから、索引ファイルであることを見落として、機械的に AT END, NOT AT END を選択してしまったと考えられる。

a~c は、追加・削除処理でレコードが存在したとき又は存在しなかったときに、それぞれどのような手順に進むかという基本的な処理であるので、ぜひ理解してもらいたい。

d では、クの誤答が多かった。これは、この PERFORM の手順の最後で、W-BANGO を1増加する処理を見落としたものと思われる。

設問2は、索引ファイルのキーの値を変更する場合にどうするかという、索引ファイルの使い方についての基本的な理解を問うものであったが、正答率が低く、キーの値を直接変更して REWRITE 文で置換するアの誤答が多かった。キー項目の値の変更は、削除・追加処理の組合せで行うという基本をぜひ理解してもらいたい。

問8

キーと値を1対1に対応付けて保持するプログラムを完成することについて出題した。このプログラムを理解するポイントは、メンバ変数 reverse の用法にあった。

四つの解答箇所 a~d のうち、a の正答率は非常に高く、よく理解されているようであった。

b は、正答率が低く、イの誤答が多かった。メンバ変数 map と reverse の宣言に着目すれば、正答することができた。

c と d は、正答率が非常に低かった。相互に関連がある解答箇所であり、メンバ変数 reverse の内容をどう変化させればよいかに着目すれば正答することができた。

この問題のような汎用的なクラスを実現するときには、新たに出题範囲となった Generics (総称) の知識が必要になるので、しっかり理解してもらいたい。

問9

2語の中にあるビット列を結合する副プログラムと1語のデータを2語に分割する副プログラムを完成することについて出題した。全体として、プログラムの処理内容はよく理解されていたようである。

設問1では、a と b の正答率が高かった。誤答として、b ではシフトの方向の誤りが、c ではレジスタの意味の取り違いによる誤りが、それぞれ見受けられた。

設問2でも、誤答としてレジスタの意味の取り違いによる誤りが見受けられた。

アセンブラのプログラムを理解するためには、各レジスタの役割をしっかりと把握することが重要である。

問 10

完成したプログラムを示し、そこから処理内容を読み取り、プログラムの実行に沿った変数の値の変化を追跡することについて出題した。

設問 1 では、a は条件が成立しない場合の処理が、b~d は条件が成立する場合の処理が、それぞれ記述されていた。a, c, d の正答率は平均的であり、b の正答率が高かった。a は“e”がプログラム中の if 文に明らかに出てこないことから、正答を見つけるのが難しかったと思われる。c, d は、どちらも if 文の条件として書かれているので、プログラムをきちんと追跡すれば正答できた。

設問 2 は、プログラムの実行結果としての返却値を問う問題であった。f の正答率が非常に低かった。条件を満足しないものが見つかって while ループを終了したとき、flag の値が 1 になっていることを理解していれば、return strlen(target) が実行されることが分かる。このとき、target のポインタがまだ進んでいないことから、正答の力を見つけることができた。

問 11

部品展開表を使って部品注文ファイルを作成するプログラムについて出題した。

四つの解答箇所 a~d のうちでは、c の正答率が低かったが、ほかは平均的であった。在庫チェックや必要部品数の計算の仕方など、プログラムの処理内容は理解されていたと考えられる。

c は START 文の KEY 句の使い方に関するもので、COBOL の索引ファイルの動的呼出しの基本である。この手続を実行する前に LOW-VALUE をキーの一部に設定していることから、誤答として多かったウの KEY = B-BUHIN-KEY の条件は成り立たない。

ファイルの呼出し法や START 文での位置付けと READ 文での順読みは、実務でもよく使われる重要な技法である。文法と使い方の両面から、十分に理解して身につけるよう心がけてもらいたい。

問 12

図書の貸出しと返却の処理のプログラムを完成することについて出題した。

設問 1 では、三つの解答箇所 a~c のうち、c の正答率は高かったが、b は低かった。

a は、イとオの誤答が多かった。初期化すべきフィールドは理解しているが、その方法を理解していないようであった。Java の文法をしっかりと身につけてもらいたい。

b は、単純にメッセージを出力する選択肢であるエとウの誤答が多かった。図に示された実行結果に注意すれば、メッセージの前に failed が出力されていることから、単純な逐次実行ではなく、メソッド main の catch ブロックへのジャンプが必要であることが分かる。例外処理の機構をしっかりと理解してほしい。

設問 2 では、e の正答率は高かった。d の正答率は平均的であったが、誤答が分散していた。メソッドの様子がどのように実現されているかを理解してほしい。

問 13

ビット列のパターンマッチングと置換について出題した。

設問 1 では、a~c の正答率が高かった。d の正答率は平均的であったが、誤答が分散していた。プログラムの流れの理解不足、及びシフト演算命令のオペランドが定数ではなくレジスタになった場合の使用方法の理解不足が原因と思われる。アセンブラによるビット処理の技法をよく理解しておいてもらいたい。

設問 2 も平均的な正答率であったが、やはり誤答が分散していた。設問 1 の穴埋め問題より正答率が低いという結果から考えると、設問 1 の穴埋めを命令の前後関係だけで判断し、プログラムの全体の流れは理解していなかった可能性がある。

アセンブラのプログラミングでは、各レジスタの役割を把握し、全体の流れを追うことによって、プログラムが何をしているかを読み取る力を身につけてもらいたい。