IT Skill Test GIC Myanmar

Duration: 30 Minutes *Total Questions: 8*

---

1. You are developing a method to calculate the total cost of items in a shopping cart. The method takes two parameters: an array of item prices (doubles) and an array of quantities (integers). Which of the following code snippets correctly calculates the total cost?

A.
```
double total = 0;
for (int i = 0; i < prices.length; i++) {
    total += prices[i] * quantities[i];
}
return total;
```

B.
```
int total = 0;
for (int i = 0; i < prices.length; i++) {
    total += prices[i] * quantities[i];
}
return total;
```

C.
```
double total = 0;
for (double price : prices) {
    total += price;
}
return total;
```

D.
```
double total = 0;
for (int i = 0; i < prices.length; i++) {
    total += prices[i] + quantities[i];
}
return total;
```

2. Which of the following logical expressions correctly checks if a given integer 'num' is divisible by both 2 and 3, but not by 5?

A. (num % 2 == 0) && (num % 3 == 0) && !(num % 5 == 0)

B. (num % 2 == 0) || (num % 3 == 0) && (num % 5 != 0)

C. (num % 2 == 0) && (num % 3 == 0) || (num % 5 != 0)

D. !(num % 2 != 0) && !(num % 3 != 0) && (num % 5 != 0)

3. You're implementing a method to find the maximum value in an array of integers. Which of the following implementations is correct and most efficient?

A.
```
int max = Integer.MIN_VALUE;
for (int num : arr) {
    if (num > max) max = num;
}
return max;
```

B.
```
int max = arr[0];
for (int i = 1; i < arr.length; i++) {
    if (arr[i] > max) max = arr[i];
}
return max;
```

C.
```
Arrays.sort(arr);
return arr[arr.length - 1];
```

D.
```
int max = 0;
for (int num : arr) {
    max = Math.max(max, num);
}
return max;
```

4. In a method that checks if a string is a palindrome, which of the following conditions correctly compares characters at opposite ends of the string?

A. s.charAt(i) == s.charAt(s.length() - i)

B. s.charAt(i) == s.charAt(s.length() - i - 1)

C. s.charAt(i) == s.charAt(s.length() - (i + 1))

D. s.charAt(i + 1) == s.charAt(s.length() - i)

5. You're implementing a binary search algorithm. Which of the following correctly calculates the middle index of the search range?

A. int mid = (low + high) / 2;

B. int mid = low + (high - low) / 2;

C. int mid = (low + high + 1) / 2;

D. int mid = (low + high) >>> 1;

6. You're developing a method to check if a year is a leap year. Which of the following logical expressions correctly implements the leap year rule (divisible by 4, but not by 100 unless also divisible by 400)?

A. (year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0)

B. (year % 4 == 0) && ((year % 100 != 0) || (year % 400 == 0))

C. (year % 4 == 0) || (year % 100 == 0) && (year % 400 == 0)

D. (year % 4 == 0) && (year % 100 == 0) && (year % 400 == 0)

7. You're implementing a method to find the longest common prefix among an array of strings. Which of the following approaches is correct and efficient?

A.
```
String prefix = strs[0];
for (int i = 1; i < strs.length; i++) {
    while (strs[i].indexOf(prefix) != 0) {
        prefix = prefix.substring(0, prefix.length() - 1);
        if (prefix.isEmpty()) return "";
    }
}
return prefix;
```

B.
```
Arrays.sort(strs);
String first = strs[0];
String last = strs[strs.length - 1];
int c = 0;
while (c < first.length() && first.charAt(c) == last.charAt(c)) c++;
return first.substring(0, c);
```

C.
```
StringBuilder result = new StringBuilder();
for (int i = 0; i < strs[0].length(); i++) {
    char c = strs[0].charAt(i);
    for (int j = 1; j < strs.length; j++) {
        if (i >= strs[j].length() || strs[j].charAt(i) != c) {
            return result.toString();
        }
    }
    result.append(c);
}
return result.toString();
```

D.
```
return strs[0].substring(0, IntStream.range(0, strs[0].length())
    .filter(i -> Arrays.stream(strs).allMatch(s -> s.length() > i && s.charAt(i) == strs[0].charAt(i)))
    .min()
    .orElse(0));
```

8. You're implementing a method to find the maximum subarray sum in an array of integers. Which of the following implementations correctly solves this problem using Kadane's algorithm?

A.
```
int maxSum = arr[0], currentSum = arr[0];
for (int i = 1; i < arr.length; i++) {
    currentSum = Math.max(arr[i], currentSum + arr[i]);
    maxSum = Math.max(maxSum, currentSum);
}
return maxSum;
```

B.
```
int maxSum = Integer.MIN_VALUE, currentSum = 0;
for (int num : arr) {
    currentSum = Math.max(num, currentSum + num);
    maxSum = Math.max(maxSum, currentSum);
}
return maxSum;
```

C.
```
int maxSum = arr[0];
for (int i = 1; i < arr.length; i++) {
    if (arr[i] > arr[i] + arr[i-1]) {
        arr[i] = arr[i];
    } else {
        arr[i] += arr[i-1];
    }
    maxSum = Math.max(maxSum, arr[i]);
}
return maxSum;
```

D.
```
return Arrays.stream(arr).reduce(0, (a, b) -> Math.max(a + b, b));
```