Accenture 2025

Accenture Coding Questions

S. No.	Problem	Java	Python

1	Calculate Dividend index in an array based on Q, D, R given	import java.util.*;
		
		public class Main {
		 public static int function(int[] arr, int Q, int D, int R) {
		  int n = arr.length;
		  int div = D * Q + R;
		  for (int i = 0; i < n; i++) {
		   if (arr[i] == div) {
		    return i;
		   }
		  }
		  return -1;
		 }
		 public static void main(String[] args) {
		  int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
		  int Q = 2, D = 2, R = 1;
		  System.out.println(function(arr, Q, D, R));
		 }
		}
			def fun(arr, D, Q, R):
			 div = (Q * D) + R
			 for i in range(len(arr)):
			  if (arr[i] == div):
			   return i
			 return -1
			def main():
			 arr = [1, 2, 3, 4, 5, 6, 7, 8]
			 D = 2
			 Q = 2
			 R = 1
			 print(fun(arr, D, Q, R))
			if __name__ == "__main__":
			 main()

2	Even position sum after reversing	static void AbsEvenOdd() {
		    int arr[] = {1,2,3,4,5,6,7,8};
		    int even = 0, fg = 1;
		    if(arr.length % 2 != 0)
		      fg = 0;
		    for(int i=fg; i<arr.length; i=i+2) {
		      even += arr[i];
		    }
		    System.out.println("Count " + even);
		  }
			def fun(arr):
			 even = 0
			 flag = 0
			 if (len(arr) % 2 == 0):
			  flag = 1
			 for i in range(flag, len(arr), 2):
			  even += arr[i]
			 return even
			def main():
			 arr = [1, 2, 3, 4, 5, 6, 7, 8]
			 print(fun(arr))
			if __name__ == "__main__":
			 main()

3	Maximum Candies you can buy	//Candies
		  static int candies(int len,int[] arr, int amount) {
		    int candy = 0;
		    Arrays.sort(arr);
		    for(int i: arr) {
		      if(i % 5 ==0) {
		        candy+=1;
		      }
		      else if(amount<i) {
		        continue;
		      }
		      else {
		        amount -= i;
		        candy+=1;
		      }
		    }
		    return candy;
		  }
			def candies(length, arr, amount):
			  candy = 0
			  arr.sort()
			  for i in arr:
			    if i % 5 == 0:
			      candy += 1
			    elif amount < i:
			      continue
			    else:
			      amount -= i
			      candy += 1
			  return candy

4	Absolute Difference of Odd index - Sum and Even index - XOR	import java.util.*;
		
		public class Main {
		 public static int function(int[] arr) {
		  int n = arr.length;
		  int odd = 0, even = arr[0];
		  for (int i = 1; i < n; i++) {
		   if (i % 2 == 0) {
		    even ^= arr[i];
		   } else {
		    odd += arr[i];
		   }
		  }
		  int abs = Math.abs(odd - even);
		  return abs;
		 }
		 public static void main(String[] args) {
		  int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
		  System.out.println(function(arr));
		 }
		}
			def fun(arr):
			 odd = 0
			 even = arr[0]
			 for i in range(1, len(arr)):
			  if (i % 2 == 0):
			   even ^= arr[i]
			  else:
			   odd += arr[i]
			 return abs(odd - even)
			def main():
			 arr = [1, 2, 3, 4, 5, 6, 7, 8]
			 print(fun(arr))
			if __name__ == "__main__":
			 main()

5	Character Count ('l' count in a String "Hello World")	//Char Count
		static int CharCount(String s, char a) {
		    int count = 0;
		    for(char c : s.toCharArray()) {
		      if(c == a)
		        count+=1;
		    }
		    return count;
		  }
			def fun(s, find):
			 count = 0
			 for i in range(len(s)):
			  if (s[i] == find):
			   count += 1
			 return -1 if count == 0 else count
			def main():
			 s = "Hello World"
			 find = "l"
			 print(fun(s, find))
			if __name__ == "__main__":
			 main()

6	Continuous Temperature Drop	//Continous Temperature Drop
		static int TemperatureDrop(int arr[]) {
		    int count = 1;
		    int max = 0;
		    for(int i=1; i<arr.length; i++) {
		      if(arr[i] < arr[i-1])
		        count+=1;
		      else
		        count = 1;
		      max = Math.max(max, count);
		    }
		    return max;
		  }
			def fun(temp):
			 cnt = 0
			 mx = 0
			 for i in range(len(temp)-1):
			  if (temp[i] > temp[i+1]):
			   cnt += 1
			  else:
			   mx = max(mx, cnt)
			   cnt = 0
			 return mx
			def main():
			 temp = [10, -5, -7, -2, 3, 2, 1, 0, -1, 5, -7, -1]
			 print(fun(temp))
			if __name__ == "__main__":
			 main()

7	Count Lowercase in a String and print with their count	//Count LowerCase
		static void LowerCase(String s) {
		    HashMap<Character, Integer> hm = new HashMap<>();
		    for(char c : s.toCharArray()) {
		      if(Character.isLowerCase(c)) {
		        hm.put(c,hm.getOrDefault(c,0)+1);
		      }
		    }
		    for(Map.Entry<Character, Integer> entry : hm.entrySet()) {
		      System.out.println(entry.getKey() + entry.getValue());
		    }
		  }
			def fun(s):
			 cnt = 0
			 res = ""
			 for i in range(len(s)):
			  if (ord(s[i]) >= 97 and ord(s[i]) <= 122):
			   res += s[i]
			   cnt += 1
			 return "{}: {}".format(res, cnt)
			def main():
			 s = "Hello WoRld"
			 print(fun(s))
			if __name__ == "__main__":
			 main()

8	Even / Odd of Absolute difference of Whitespaces in two Strings	public static int fun(String s1, String s2) {
		  int cnt1 = 0, cnt2 = 0;
		  for (int i = 0; i < s1.length(); i++) {
		    if (s1.charAt(i) == ' ') {
		      cnt1++;
		    }
		  }
		  for (int i = 0; i < s2.length(); i++) {
		    if (s2.charAt(i) == ' ') {
		      cnt2++;
		    }
		  }
		  return Math.abs(cnt1 - cnt2);
		}
			def fun(s1, s2):
			  cnt1, cnt2 = 0, 0
			  for i in range(len(s1)):
			    if (s1[i] == " "):
			      cnt1 += 1
			  for i in range(len(s2)):
			    if (s2[i] == " "):
			      cnt2 += 1
			  return abs(cnt1 - cnt2)
			def main():
			  s1 = "Hello WoRld"
			  s2 = "Welcome to Programming of Python "
			  print(fun(s1, s2))
			if __name__ == "__main__":
			  main()

9	Factorial of absolute difference of (Vowel count - Length of String)	public static int fun(String s) {
		    char[] vowels = {'a', 'e', 'i', 'o', 'u'};
		    int cnt = 0;
		    int fact = 1;
		    for (int i = 0; i < s.length(); i++) {
		      char c = s.charAt(i);
		      boolean isVowel = false;
		      for (char vowel : vowels) {
		        if (c == vowel) {
		          isVowel = true;
		          break;
		        }
		      }
		      if (!isVowel && c != ' ') {
		        cnt++;
		        fact *= cnt;
		      }
		    }
		    return fact;
		  }
			def fun(s):
			  vowels = ['a', 'e', 'i', 'o', 'u']
			  cnt = 0
			  fact = 1
			  for i in range(len(s)):
			    if s[i] not in vowels and s[i] != " ":
			      cnt += 1
			      fact *= cnt
			  return fact
			def main():
			  s = "Hello wor"
			  print(fun(s))
			if __name__ == "__main__":
			  main()

10	Height of the N-th bounce	static double formulae(double h, double v,double vn) {
		    double en = v/vn;
		    double hn = h * (Math.pow(en,2));
		    return hn;
		  }
			def formulae(h, v, vn):
			    en = v / vn
			    hn = h * (en ** 2)
			    return hn
			def main():
			    h = 10.0
			    v = 5.0
			    vn = 2.0
			    print(formulae(h, v, vn))
			if __name__ == "__main__":
			    main()

11	Highest Product with first number max for Two Sum	public static int[] fun(int[] arr, int target) {
		    int mx = 0;
		    int[] res = new int[2];
		    int val = 0;
		    for (int i = 0; i < arr.length; i++) {
		      int complement = target - arr[i];
		      for (int j = 0; j < arr.length; j++) {
		        if (arr[j] == complement) {
		          val = arr[i] * complement;
		          break;
		        }
		      }
		      if (val > mx) {
		        mx = val;
		        res[0] = Math.max(arr[i], complement);
		        res[1] = Math.min(arr[i], complement);
		      }
		    }
		    return res;
		  }
			def fun(arr, target):
			  mx = 0
			  res = [0] * 2
			  val = 0
			  for i in range(len(arr)):
			    if (target - arr[i]) in arr:
			      val = arr[i] * (target - arr[i])
			    if (val > mx):
			      mx = max(mx, val)
			      res[0] = max(arr[i], (target - arr[i]))
			      res[1] = min(arr[i], (target - arr[i]))
			  return res
			def main():
			  arr = [1, 2, 3, 4, 5, 6, 7]
			  target = 7
			  print(fun(arr, target))
			if __name__ == "__main__":
			  main()

12	Highest version of a file in a String Array (Exception Handle)	static int FileHandling(String[] files) {
		    try {
		        int val = 0;
		        for(String s : files) {
		            val = Math.max(Integer.valueOf(s.replace("file_","")),val);
		        }
		        return val;
		    }
		    catch(Exception e) {
		        return -1;
		    }
		}
			def file_handling(files):
			    try:
			        val = 0
			        for s in files:
			            num = int(s.replace("file_", ""))
			            val = max(num, val)
			        return val
			    except Exception:
			        return -1
			def main():
			    files = ["file_1", "file_20", "file_3", "file_10"]
			    print(file_handling(files))
			if __name__ == "__main__":
			    main()

13	Life on Island	//Life on Island
		    static int LifeOnIsland(int choki, int limit, int days) {
		        if(choki == 0)
		            return -1;
		        int tot = limit * days;
		        int count = 0;
		        while(tot>0) {
		            tot -= choki;
		            count += 1;
		        }
		        return count;
		    }
			def life_on_island(choki, limit, days):
			  if choki == 0:
			     return -1
			  tot = limit * days
			  count = 0
			  while tot > 0:
			     tot -= choki
			     count += 1
			  return count
			def main():
			  choki = 5
			  limit = 10
			  days = 3
			  print(life_on_island(choki, limit, days))
			if __name__ == "__main__":
			  main()

14	Longest length of the uninterrupted String
	Alice number (123.45.6789.0 return 4 maximum length between two "." = 4)
		//Alice Number
		    static int LongestLength(String s) {
		        if(s.indexOf('.') == -1)
		            return 0;
		        int max = 0;
		        String arr[] = s.split("\\.");
		        for(String token : arr) {
		            max = Math.max(token.length() , max);
		        }
		        return max;
		    }
			def longest_length(s):
			  if '.' not in s:
			     return 0
			  arr = s.split('.')
			  max_length = 0
			  for token in arr:
			     max_length = max(len(token), max_length)
			  return max_length
			def main():
			  s = "123.456.7890.12"
			  print(longest_length(s))
			if __name__ == "__main__":
			  main()

15	Maximum Substring matching with a target String	//Max Substring
		static int max(String s, String arr[]) {
		        int max = 0;
		        for(String i : arr) {
		            if(s.contains(i)) {
		                max = Math.max(max,i.length());
		            }
		        }
		        return max;
		    }
			def max_substring(s, arr):
			  max_length = 0
			  for substring in arr:
			     if substring in s:
			        max_length = max(max_length, len(substring))
			  return max_length
			def main():
			  s = "hello world"
			  arr = ["hello", "world", "lo", "worl"]
			  print(max_substring(s, arr))
			if __name__ == "__main__":
			  main()

16	Odd or Even - return as a String	public static String fun(int n) {
		    StringBuilder res = new StringBuilder();
		    while (n > 0) {
		        if ((n % 10) % 2 == 0) {
		            res.insert(0, "Even");
		        } else {
		            res.insert(0, "Odd");
		        }
		        n /= 10;
		    }
		    return res.toString();
		}
			def fun(n):
			  res = ""
			  while (n > 0):
			    if ((n % 10) % 2 == 0):
			      res = "Even" + res
			    else:
			      res = "Odd" + res
			    n //= 10
			  return res
			def main():
			  n = 234566
			  print(fun(n))
			if __name__ == "__main__":
			  main()

17	Permutation (Consonants possibilities with fixed vowels)	public static int fun(String s) {
		    char[] vowels = {'a', 'e', 'i', 'o', 'u'};
		    int cnt = 0;
		    int fact = 1;
		    for (int i = 0; i < s.length(); i++) {
		        char c = s.charAt(i);
		        boolean isVowel = false;
		        for (char vowel : vowels) {
		            if (c == vowel) {
		                isVowel = true;
		                break;
		            }
		        }
		        if (!isVowel && c != ' ') {
		            cnt++;
		            fact *= cnt;
		        }
		    }
		    return fact;
		}
			def fun(s):
			  vowels = ['a', 'e', 'i', 'o', 'u']
			  cnt = 0
			  fact = 1
			  for i in range(len(s)):
			    if s[i] not in vowels and s[i] != " ":
			      cnt += 1
			      fact *= cnt
			  return fact
			def main():
			  s = "Hello wor"
			  print(fun(s))
			if __name__ == "__main__":
			  main()

18	Round off - Calculate (Formula - given)	static int roundOff(int l, int b) {
		    int res = (int)Math.round((3.14 * l * b));
		    return res;
		}
			import math
			def round_off(l, b):
			    res = round(3.14 * l * b)
			    return res

19	Second Largest Repeating Number	static int SecondLargest(int arr[]) {
		    HashMap<Integer, Integer> hm = new HashMap<>();
		    HashSet<Integer> hs = new HashSet<>();
		    int min1 = Integer.MIN_VALUE, min2 = Integer.MIN_VALUE;
		    for(int i: arr) {
		        hm.put(i,hm.getOrDefault(i,0)+1);
		        if(!hs.add(i)) {
		            if(i>min1) {
		                min2 = min1;
		                min1 = i;
		            }
		            else if (i>min2 && min1 != i) {
		                min2 = i;
		            }
		        }
		    }
		    return hm.get(min2);
		}
			def second_largest(arr):
			  from collections import defaultdict
			  counts = defaultdict(int)
			  seen = set()
			  min1 = float('-inf')
			  min2 = float('-inf')
			  for i in arr:
			    counts[i] += 1
			    if i in seen:
			      if i > min1:
			        min2 = min1
			        min1 = i
			      elif i > min2 and i != min1:
			        min2 = i
			    seen.add(i)
			  return counts.get(min2, 0)
			def main():
			  arr = [4, 2, 7, 7, 5, 6, 6, 6]
			  print(second_largest(arr))
			if __name__ == "__main__":
			  main()

<table>
<tr><td>20</td><td>Special Fibonacci Series (F(n) = F(n-1) * F(n-1) + F(n-2) * F(n-2))</td><td>//FibbNoacci
  static int SplFnb(int n) {
    if(n == 0 | | n == 1)
      return n;
    return SplFnb(n-1) * SplFnb(n-1) + SplFnb(n-2) * SplFnb(n-2);
  }</td><td>def fun(n):
 if (n == 0 or n == 1):
  return n
 return ((fun(n-1) * fun(n-1)) + (fun(n-2) + fun(n-2)))
def main():
 print(fun(3))
if __name__ == "__main__":
 main()</td></tr>
</table>