

KENDRIYA VIDYALAYA SANGATHAN, JAIPUR REGION

FIRST PRE-BOARD EXAMINATION 2020-21

Class: XII Sub: Computer Science (083)

MARKING SCHEME

Maximum Marks: 70

Time Allowed: 3 hours

General Instructions:

1. This question paper contains two parts A and B. Each part is compulsory.
2. Both Part A and Part B have choices.
3. Part-A has 2 sections:
 - a. Section – I is short answer questions, to be answered in one word or one line.
 - b. Section – II has two case studies questions. Each case study has 4 case-based subparts. An examinee is to attempt any 4 out of the 5 subparts.
4. Part - B is Descriptive Paper.
5. Part- B has three sections
 - a. Section-I is short answer questions of 2 marks each in which two question have internal options.
 - b. Section-II is long answer questions of 3 marks each in which two questions have internal options.
 - c. Section-III is very long answer questions of 5 marks each in which one question has internal option.
6. All programming questions are to be answered using Python Language only

| Q. No. | Part-A | Marks |
|--------|--|-------|
| | Section- I Select the most appropriate option out of the options given for each question. Attempt any 15 questions from question no 1 to 21. | |
| 1 | c) 3rdPlace (1 Mark for correct answer, No partial marking) | 1 |
| 2 | [2, 55, 87,1] (1 Mark for correct answer, ½ mark for partial correct answer) | 1 |
| 3 | pickle (1 Mark for correct answer, No partial marking) | 1 |
| 4 | d) <> (1 Mark for correct answer, No partial marking) | 1 |

| | | |
|----|---|---|
| 5 | b) print(L[-1]) (1 Mark for correct answer, No partial marking) | 1 |
| 6 | country={"india":"New Delhi","Sri Lanka":"Colombo","China":"Beijing"} or any other correct answer. (1 Mark for correct answer, ½ mark for partial correct answer) | 1 |
| 7 | Anil (1 Mark for correct answer, No partial marking) | 1 |
| 8 | capitalize() (1 Mark for correct answer, No partial marking) | 1 |
| 9 | SMTP (1 Mark for correct answer, No partial marking) | 1 |
| 10 | Cyberstalking or cyberbullying (1 Mark for correct answer, No partial marking) | 1 |
| 11 | DISTINCT (1 Mark for correct answer, No partial marking) | 1 |
| 12 | BETWEEN command/clause is used to retrieve values within a range in a SELECT, INSERT, UPDATE, or DELETE statement. (1 Mark for correct answer, ½ mark for partial correct answer) | 1 |
| 13 | SUM(),count() or any other two correct answer. (½ mark for each correct answer) | 1 |
| 14 | INSERT (1 Mark for correct answer, NO partial marking) | 1 |
| 15 | Radio wave or Microwave or Satellite (1 Mark for correct answer, NO partial marking) | 1 |
| 16 | b. string (1 Mark for correct answer, NO partial marking) | 1 |

| | | |
|----|--|---|
| 17 | Basic (1 Mark for correct answer, ½ mark for partial correct answer) | 1 |
| 18 | SELECT COUNT(*) FROM EMP; (1 Mark for correct answer, ½ mark for partial correct answer) | 1 |
| 19 | Uniform Resource Locator. (1 Mark for correct answer, ½ mark for partial correct answer) | 1 |
| 20 | b) One (1 Mark for correct answer, NO partial marking) | 1 |
| 21 | bps,Kbps,Mbps,Gbps,Tbps (1 Mark for correct answer, NO partial marking) | 1 |

| | | |
|----|---|---|
| 23 | <p>Suresh is writing a program to create a CSV file “files.csv” which will contain filetypes and file extensions for some records. He has written the following code. As a programmer, help him to successfully execute the given task.</p> <pre> Import ----- # Line 1 def addinFile(filetype,extension): # to write /add data into the file f=open('-----','-----') # Line 2 newFileWriter = csv.writer(f) newFileWriter.writerow([filetype,extension]) f.close() #csv file reading code def readfile(filename): # to read data from CSV with open(filename,'r') as nf: nfr = csv._____(nf) # Line 3 for row in nfr: print (row[0],row[1]) nf._____ # Line 4 addinFile(“C++”, “.cpp”) addinFile(“Python”, “.py”) addinFile(“Java”, “.java”) addinFile(“Microsoft Excel”, “.xls”) readfile(.....) #Line 5 </pre> | |
| | (a) import csv | 1 |
| | (b) a | 1 |
| | (c) reader | 1 |
| | (d) close() | 1 |
| | (e) files.csv | 1 |
| | (1 Mark for each correct answer, NO partial marking) | |
| | Part – B | |
| | Section-I | |

| 24 | <table border="1"> <thead> <tr> <th data-bbox="304 136 373 232">Sr. no.</th> <th data-bbox="373 136 509 232">Expression</th> <th data-bbox="509 136 758 232">Operation</th> <th data-bbox="758 136 928 232">Stack</th> <th data-bbox="928 136 1350 232">Postfix</th> </tr> </thead> <tbody> <tr><td>1</td><td>P</td><td>Append</td><td></td><td>P</td></tr> <tr><td>2</td><td>-</td><td>PUSH</td><td>-</td><td>P</td></tr> <tr><td>3</td><td>Q</td><td>Append</td><td>-</td><td>P Q</td></tr> <tr><td>4</td><td>*</td><td>PUSH</td><td>-*</td><td>P Q</td></tr> <tr><td>5</td><td>R</td><td>Append</td><td>-*</td><td>P Q R</td></tr> <tr><td>6</td><td>/</td><td>POP,PUSH</td><td>-/</td><td>P Q R *</td></tr> <tr><td>7</td><td>(</td><td>PUSH</td><td>-/(</td><td>P Q R *</td></tr> <tr><td>8</td><td>S</td><td>Append</td><td>-/(</td><td>P Q R * S</td></tr> <tr><td>9</td><td>+</td><td>PUSH</td><td>-/(+</td><td>P Q R * S</td></tr> <tr><td>10</td><td>(</td><td>PUSH</td><td>-/(+(</td><td>P Q R * S</td></tr> <tr><td>11</td><td>T</td><td>Append</td><td>-/(+(</td><td>P Q R * S T</td></tr> <tr><td>12</td><td>+</td><td>PUSH</td><td>-/(+(+</td><td>P Q R * S T</td></tr> <tr><td>13</td><td>U</td><td>Append</td><td>-/(+(+</td><td>P Q R * S T U</td></tr> <tr><td>14</td><td>)</td><td>POP</td><td>-/(+</td><td>P Q R * S T U +</td></tr> <tr><td>15</td><td>)</td><td>POP</td><td>-/</td><td>P Q R * S T U + +</td></tr> <tr><td>16</td><td>*</td><td>POP,PUSH</td><td>-*</td><td>P Q R * S T U + + /</td></tr> <tr><td>17</td><td>V</td><td>Append</td><td>-*</td><td>P Q R * S T U + + /</td></tr> <tr><td>18</td><td>EMPTY</td><td>Pop all</td><td></td><td>P Q R * S T U + + / * -</td></tr> </tbody> </table> <p data-bbox="304 1406 1310 1442">(2 Mark for correct answer, deduct ½ mark for each wrong conversion)</p> | Sr. no. | Expression | Operation | Stack | Postfix | 1 | P | Append | | P | 2 | - | PUSH | - | P | 3 | Q | Append | - | P Q | 4 | * | PUSH | -* | P Q | 5 | R | Append | -* | P Q R | 6 | / | POP,PUSH | -/ | P Q R * | 7 | (| PUSH | -/(| P Q R * | 8 | S | Append | -/(| P Q R * S | 9 | + | PUSH | -/(+ | P Q R * S | 10 | (| PUSH | -/(+(| P Q R * S | 11 | T | Append | -/(+(| P Q R * S T | 12 | + | PUSH | -/(+(+ | P Q R * S T | 13 | U | Append | -/(+(+ | P Q R * S T U | 14 |) | POP | -/(+ | P Q R * S T U + | 15 |) | POP | -/ | P Q R * S T U + + | 16 | * | POP,PUSH | -* | P Q R * S T U + + / | 17 | V | Append | -* | P Q R * S T U + + / | 18 | EMPTY | Pop all | | P Q R * S T U + + / * - | 2 |
|---------|---|-----------|------------|-------------------------|-------|---------|---|---|--------|--|---|---|---|------|---|---|---|---|--------|---|-----|---|---|------|----|-----|---|---|--------|----|-------|---|---|----------|----|---------|---|---|------|-----|---------|---|---|--------|-----|-----------|---|---|------|------|-----------|----|---|------|-------|-----------|----|---|--------|-------|-------------|----|---|------|--------|-------------|----|---|--------|--------|---------------|----|---|-----|------|-----------------|----|---|-----|----|-------------------|----|---|----------|----|---------------------|----|---|--------|----|---------------------|----|-------|---------|--|-------------------------|---|
| Sr. no. | Expression | Operation | Stack | Postfix | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | P | Append | | P | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | - | PUSH | - | P | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Q | Append | - | P Q | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | * | PUSH | -* | P Q | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | R | Append | -* | P Q R | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | / | POP,PUSH | -/ | P Q R * | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | (| PUSH | -/(| P Q R * | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | S | Append | -/(| P Q R * S | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9 | + | PUSH | -/(+ | P Q R * S | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | (| PUSH | -/(+(| P Q R * S | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | T | Append | -/(+(| P Q R * S T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | + | PUSH | -/(+(+ | P Q R * S T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 13 | U | Append | -/(+(+ | P Q R * S T U | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 14 |) | POP | -/(+ | P Q R * S T U + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 |) | POP | -/ | P Q R * S T U + + | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | * | POP,PUSH | -* | P Q R * S T U + + / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | V | Append | -* | P Q R * S T U + + / | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 18 | EMPTY | Pop all | | P Q R * S T U + + / * - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | (2 Mark for correct differences, 1 mark for partial correct answer) | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 26 | <p data-bbox="469 1648 1142 1794"> a. https :hypertext transfer protocol secure b. VoIP : Voice over Internet Protocol c. GPRS : Global system for mobile communication d. IPR : Intellectual property rights </p> <p data-bbox="304 1827 807 1863">(1/2 Mark for each correct answer)</p> | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | |
|----|--|---|
| 27 | <p>(1 mark for correct difference and 1 mark for correct example) or (1/2 mark for partial correct difference and 1/2 mark for partial correct example)</p> <p style="text-align: center;">OR</p> <pre>x= 100 # Global var x def myfunc(a): # Local var a k=a # Local var k print(k,a) p=(0,1,2,3,4) # Global var p myfunc(p) print(x)</pre> <p>(1/2 mark for each correct answer)</p> | 2 |
| 28 | <p><u>x=310</u></p> <pre>for z in range(0,x) : if z//5==0: print (z**5) elif z%5==0: print (z+300)</pre> <p>(1/2 mark for each correct answer)</p> | 2 |
| 29 | <p>Option (i),(iii) &(iv) (1½ mark for correct answer)(No partial marking) Maximum = 4 Minimum=1 (½ mark for correct answer)</p> | 2 |
| 30 | <p>(1 mark for correct definition and 1 mark for correct example) (½ mark for partial correct definition and ½ mark for partial correct example)</p> | 2 |
| 31 | <p>(1 mark for correct definition and 1 mark for correct example) (½ mark for partial correct definition and ½ mark for partial correct example)</p> | 2 |
| 32 | <p>Write the full forms of DDL and DML. Write any one command of each. (½ mark for each correct full form and ½ mark for each correct example) (NO mark for partial correct form (if any))</p> | 2 |
| 33 | <p>C*@@@K*#@@</p> <p>(2 mark for correct answer) or (1½ or 1 or ½ mark for partial correct answer)</p> | 2 |
| | <p>Section- II</p> | |

| | | |
|----|---|---|
| 34 | <pre>def Display(num): for i in num: if i%10==0: print(i,end=" ") print() for i in num: if i%10!=0: print(i,end=" ")</pre> <p><i>(½ marks for correct function header)</i> <i>(1 mark for correct loop for printing number ending with 0)</i> <i>(1 mark for correct loop for printing number not ending with 0)</i> <i>(½ mark for printing number in respective line)</i></p> | 3 |
| 35 | <pre>def DisplayWords(): f=open("D://STORY.txt","r") s=f.read() for w in s.split(): if "s" in w or "S" in w: print(w) f.close() OR def WordCount(): f=open("D://mydiary.txt","r") ln=0 for line in f: ln=ln+1 c=0 for word in line.split(): c=c+1 print("Line No",ln,":",c) f.close()</pre> <p><i>(½ marks for correct function header)</i> <i>(½ mark for correct opening file)</i> <i>(½ mark for correct reading from file)</i> <i>(½ mark for correct condition or counting loop)</i> <i>(½ mark for printing output correctly)</i> <i>(½ mark for closing file correctly)</i></p> | 3 |

- i) General 2
 Ortho 2
 ENT 2
 Heart 2
- ii) 400000 120000 31/07/2018
- iii) Parveen Ortho Jodhpur
 Satyajeet ENT Jaipur
 Vijay ENT Jaipur
 Kamlesh Ortho Jodhpur

(1 mark for each correct answer)

($\frac{1}{2}$ mark for each partially answer)


```

top=-1

stk=[]

def PUSH_IN(L):          # Allow additions to the stack
    for i in L:
        if i%2==0:
            stk.append(i)
            top=len(stk)-1

```

(½ marks for correct function header)

(1 mark for correct accessing of list elements)

(½ mark for correct condition for even number)

(½ mark for applying append() correctly)

(½ mark for assignment in variable top)

OR

```

def isEmpty(stk):        # checks whether the stack is empty or not
    if stk==[]:
        return True
    else:
        return False
def POP_OUT(stk):
    if isEmpty(stk):    # verifies whether the stack is empty or not
        print("Stack Underflow")
    else:               # Allow deletions from the stack
        item=stk.pop()
        if len(stk)==0:
            top=-1
        else:
            top=len(stk)
        return item

```

(½ marks for correct POP_OUT() function header)

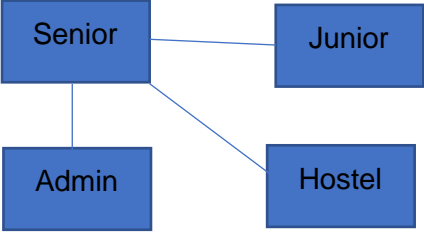
(½ mark for checking empty stack status)

(½ mark for removing item for stack)

(1 mark for assignment in variable top)

(½ mark for returning the deleted item)

Section-III

| | | |
|----|--|---|
| 38 | <p>a. Best wired medium- Twisted pair cable</p>  <p>(½ mark for correct wire medium and ½ mark for correct cable layout)</p> <p>b. The server should be installed at Wing S(Senior) as per 80-20 rule i.e. maximum traffic should be local and minimum traffic should pass over backbone. (½ mark for correct server block and ½ mark for correct justification)</p> <p>c. Firewall. (1 mark for correct answer, No partial marking)</p> <p>d. Device: Wireless Access Point or Router or WiFi hotspot device or Wifi Dongle Protocol: IEEE 802.11x or TCP/IP (½ mark for correct Device and ½ mark for correct protocol)</p> <p>e. Switch (1 mark for correct answer, No partial marking)</p> | 5 |
| 39 | <p>a. SELECT * FROM DOCTOR WHERE DEPARTMENT='ENT';</p> <p>b. SELECT DNAME FROM DOCTOR WHERE GENDER='M' AND DEPARTMENT='GENERAL' AND SALARY>120000;</p> <p>c. SELECT DNAME,DATE_OF_JOIN FROM DOCTOR ORDER BY DATE_OF_JOIN DESC;</p> <p>d. SELECT DNAME,SALARY,AGE FROM DOCTOR WHERE GENDER='F';</p> <p>e. SELECT DEPARTMENT,COUNT(*) AS "NO OF DOCTORS" FROM DOCTOR GROUP BY DEPARTMENT;</p> <p>(1 mark for each correct answer) (½ mark for each partially correct answer)</p> | 5 |
| 40 | <p>(i)</p> <pre>import pickle def CreatePC(): lst=[2726,"Accer","SHWH",49500.25] #list f=open("d:\\Computers.dat","ab+") # file in binary mode pickle.dump(lst,f) # adding in binary file f.close()</pre> <p>(½ mark for correct function header) (½ mark for correct opening of file) (½ mark for correct writing into file) (½ mark for correct file closing statement)</p> <p>(ii)</p> <pre>def FindPCs(prc): f=open("d:\\Computers.dat","rb") # Open in read mode while True: try: rec=pickle.load(f) # Reading from file till end if prc>=rec[3]: print("-----") print("CNo:",rec[0])</pre> | 5 |

```

        print("Make:",rec[1])
        print("Model:",rec[2])
        print("Price:",rec[3])
    except EOFError:
        break
    print("-----")
    f.close()
( ½ mark for correct function header)
( ½ mark for correct opening of file)
( ½ mark for correct loop for reading from file till EOF)
( ½ mark for correct reading from file)
( ½ mark for correct comparison/if condition)
( ½ mark for correct printing of record)

```

OR

```

def Player_Count():
    f=open("d:\\Club.dat","rb") # Open in read mode
    count=0
    while True:
        try:
            rec=pickle.load(f) # Reading from file till end
            if 7500>=rec[3]:
                print("-----")
                print("PNo:",rec[0])
                print("Name:",rec[1])
                print("Game:",rec[2])
                print("Fee:",rec[3])
            if rec[3]>10000:
                count=count+1
        except EOFError:
            break
    print("-----")
    print("No of player paying fee above 10000=",count)
    f.close()
( ½ mark for correct function header)
( ½ mark for correct opening of file)
( ½ mark for correct loop for reading from file till EOF)
( ½ mark for correct reading from file)
( ½ mark for correct comparison/if condition for fee 7500)
( ½ mark for correct printing of record details)
( ½ mark for correct comparison/if condition for fee 10000)
( ½ mark for correct counting)
( ½ mark for correct printing of counted records for fee 10000)
( ½ mark for correct closing of file).

```