B.Sc. (Honours) Examination, 2018

Semester–III (CBCS)

Physics

Skill Enhancing Compulsory Course: SECC-1

(Computational Physics Skills)

Time: Two Hours **Full Marks: 20**

Questions are of value as indicated in the margin.

Unless otherwise specified symbols carry their usual meanings.

Answer ***both*** questions.

1. Answer any two questions.

(a) *You are throwing a perfect dice that has six sides which are marked 1, 2, ... 6. You are counting N_1 (the number of times you get 1), N_2 (the number of times you get 2), ... N_6. After throwing the dice 1000 times you want to find out N_1, N_2, ... N_6.*
Write the outline of a code to generate 1000 pseudo-random numbers to simulate the experiment on a computer. (5)

(b) Prepare a .tex file that will produce the following after processing.

Introduction to LaTeX

Latex is the favourite tool for many scientists for preparing scientific documents. Let's start at the very beginning. You first prepare a text file with .tex extension. This file is processed to produce a .ps, or, .pdf file whichcan be viewed or printed.

It is easy to write complex equations, prepare tables for presenting data and include figures in a document. It is easy to use and elegant, too. Here is an example of an equation and how to refer it elsewhere in the text.

$$\frac{sinA}{a} = \frac{sinB}{b} = \frac{sinC}{c}$$

where **A**, **B**, and **C** represent the angles of a triangle and **a**, **c**, and **c** represent the lengths of the respective sides.

Trigger name	Energy range (GeV)
DiPFJetAve60	73 - 93
DiPFJetAve80	93 - 165
DiPFJetAve140	165 - 225

(5)

Table 1: Dijet trigger names and corresponding energy ranges used in this analysis are shown.

While equation 1 represents a *wellknown property of a triangle*, table 1 represents data from an analysis of proton-proton collisions at the **Large Hadron Collider**, CERN.

(c) Consider a file `sort.c` that contains the following code. (5)

```
#include<stdio.h>

int main()

{
```

```
  int ndata;

  // read the number of points to be fitted from a file
  FILE *infile, *outfile;
  float num[100], temp[100], x;
  int i, j;

  infile   = fopen("sort.inp", "rt");
  outfile  = fopen("sort.out", "wt");

  fscanf(infile, "%d", &ndata);
  printf ("\nNumbers to be sorted\t %d\n", ndata);
  fprintf(outfile, "\nNumbers to be sorted\t %d\n", ndata);

  for(i = 0; i < ndata; ++i)
    {
      fscanf(infile,"%f", &temp[i]);
      num[i] = temp[i];
    }

  for(i=0; i<ndata-1; ++i)
    {
      for(j=i+1; j<ndata; ++j)
{
  if(num[j]<num[i])
    {
      x = num[i];
      num[i] = num[j];
      num[j] = x;
    }
}
    }

  printf("\nGiven and ordered set of numbers --------------- \n");
  fprintf(outfile, "\nGiven and ordered set of numbers --------------- \n");
  for(i = 0; i < ndata; ++i)
    {
      printf("%d\t%f\t%f\n", i, temp[i], num[i]);
      fprintf(outfile, "%d\t%f\t%f\n", i, temp[i], num[i]);
    }

  return 0;
}
```

And the content of the sort.inp file is given below.

```
5
21.8
5.66
```

```
-2.0
3.0
18.2
```

How will you compile the code and execute it? Write the content of the sort.out file.

2. Answer any two:

   (a) Study the CLIPS code given below. Some lines are marked with line numbers. Explain clearly what each numbered line does. Which logic function does this code implement? $(0.25 \times 16 + 1 = 5)$

```
1  ; (assert (logic levels to get marks!))
2  (deffacts logic_start_facts
(logic_A TRUE)
(logic_B TRUE)
(logic_out FALSE))

(defrule rule1
3  (logic_A FALSE)
4  ?y <- (logic_out FALSE)
=>
5  (retract ?y)
6  (assert (logic_out TRUE)))

(defrule rule2
(logic_B FALSE)
?y <- (logic_out FALSE)
=>
(retract ?y)
(assert (logic_out TRUE)))

(defrule rule3
7  (logic_A TRUE)
8  (logic_B TRUE)
9  ?y <- (logic_out TRUE)
=>
10 (retract ?y)
11 (assert (logic_out FALSE)))

(defrule rule4
12 ?k <- (change logic_A to ?x)
13 ?z <- (logic_A ?)
=>
14 (retract ?k)
15 (retract ?z)
16 (assert (logic_A ?x)))

(defrule rule5
?k <- (change logic_B to ?x)
?z <- (logic_B ?)
=>
(retract ?k)
(retract ?z)
(assert (logic_B ?x)))
```

(b) i. Two tables are given below, named 'student_details' and 'tutorial_marks'. Create SQL queries (view creation preferred) to get the tutorial marks for the student Kingshuk Mandal. The output should include the student's name, tutorial_name, full_marks and marks obtained. (2.5)

student_details

id	name	semester
420	Bhutum Sarkar	40
-50	Kingshuk Mandal	-3
840	Halum Bagha	300

tutorial_marks

student_id	tutorial_name	full_marks	marks_obtained
840	Ghar Motkano	10	6
-50	Math Methods 1	10	-3
420	Bhoy Dekhano	15	14

ii. Study the following HTML code and create a diagram showing the output as will be produced in a standard bowser (the picture may be as per you artistic preferences):(2.5)

```
<html>
<body bgcolor=white text=black>
<div align=center>
<h1>On the Theory of Tittlebats</h1>
<h3><i>Charles Dickens</i></h3>
</div>
The theory of Tittlebats was first described in the book '<i><b>The Pickwick
Papers</b></i>'.
<p>
Tittlebats live in trees ...<br>
<div align=center>
<img src=tree.jpg>
</div>
As a matter of fact Tittlebats would have lived in trees if they had
satisfied the following requirements:
<p>
<ol type=A>
<li>
They should be alive when documented.
<li>
They should agree to be documented.
<li>
They should exist.
</ol>
Since the third criterion was never satisfied, no successful Ph.D.
was ever produced from this grand study.
</body>
</html>
```

(c) i. Write a C programme which will ask for a string prefix (something like `bhoot_`) and use `sprintf` to create ten names of the form <prefix>< number> (eg. `bhoot_0`, `bhoot_1`, .. , `bhoot_9`) and open ten files having these names. (2)

ii. Write CLIPS code which will factorise a single occurrence of the form $a^2 - b^2$ into the form $(a + b)(a - b)$ in an asserted line (e.g. $bhoot^2 - poot^2$ will become $(bhoot + poot)(bhoot - poot)$). (2)

iii. Write HTML code to produce a small table. (1)