Department of Computer Science and Engineering
Indian Institute of Technology Patna

Course:CS102 End Sem Exam
Time: 3 Hours Full Marks 100

Roll No: _____ Name: _____

Answer in the question paper itself. Rough work can be done in the supplementary sheets. Clearly state any reasonable assumptions you make.

1. Please tick on only one option. 10x3=30 Marks

(a) What value does the call strFunc1("PDS 2017") return?

```
int strFunc1 ( char A[] )
{
  int i = 0;
  while (A[i]) i = i + 1;
  return i;
}
```

i. 7
ii. 8
iii. 9
iv. 10

(b) What string does the following program print?

```
#include <stdio.h>
#include <string.h>
void strFunc2 ( char A[] , int n )
{
  char t;
  if (n <= 1) return;
  t = A[0]; A[0] = A[n-1]; A[n-1] = t;
  strFunc2(&A[1],n-2);
}
int main ()
{
  char A[10] = "PDS 2017";
  strFunc2(A,strlen(A));
  printf("%s", A);
}
```

i. 7012 PDS
ii. 2017 PDS
iii. 7102 SDP
iv. SDP 7102

(c) Consider the following sequence of push and pop operations on an initially empty stack S.
S = push(S,1);
S = push(S,2);
S = pop(S);
S = push(S,3);
S = push(S,4);
S = pop(S);
S = pop(S);
S = pop(S);
Which of the following is the correct order in which elements are popped?

i. 1,2,3,4
ii. 2,1,3,4
iii. 2,3,4,1
iv. 2,4,3,1

(d) The call recFunc("programming") is made, where recFunc() is defined as follows.

```
void recFunc(char A[]){
   int t;
   t = strlen(A);
   if (t == 0) return;
   else if (t % 2 == 0) recFunc(&A[t/2]);
        else recFunc(&A[1]);
}
```

How many times is recFunc() called? Include the outermost call of recFunc("programming") in the count.

i. 7
ii. 11
iii. 6
iv. 10

(e) The code below dynamically allocates space for a 2-D array:

```
double **arr;
arr = (double **) malloc(10*sizeof(double *));
for (i = 0; i < 10; i++)
  arr[i] = (double *) malloc(5*sizeof(double));
```

How many values are stored in this array, and how are they organized?

i. 15 values, organized as one array of 10 values and a separate array of 5 values.
ii. 50 values, organized as an array with 10 rows and 5 columns.
iii. 50 values, organized as an array with 5 rows and 10 columns.
iv. 100 values, organized as an array with 10 rows and 10 columns.

(f) The inorder and preorder traversal of a binary tree are d b e a f c g and a b d e c f g, respectively. The postorder traversal of the binary tree is:

i. d e b f g c a
ii. e d b g f c a
iii. e d b f g c a
iv. d e f g b c a

(g) What is the output of the following program?

```
struct node{
int cval;
struct node *next;
}
main()
{
 struct node N1, N2, N3;
 N1.cval = 1; N2.cval = 10; N3.cval = 100;
 N1.next = &N2; N2.next = &N3; N3.next = &N1;
 printf("%d,%d", N2.next->cval,
  N2.next->next->cval);
}
```

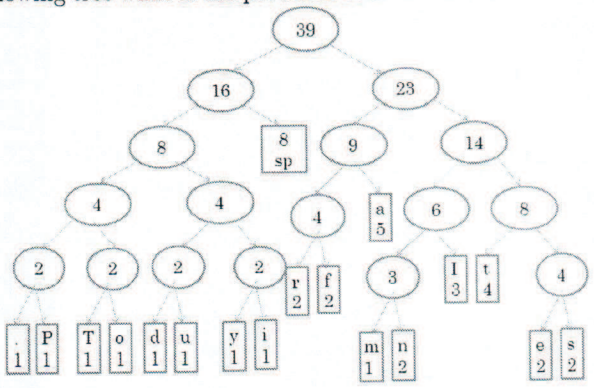
i. 1,10
ii. 1,100
iii. 10,100
iv. 100,1

(h) What is the value of s after the termination of the following loop?

```
int n = 100, s;
for (s = 2; n > 2; - -n) {
s += 2; n -= 2;
}
```

i. 64
ii. 66
iii. 68
iv. 98

(i) Following Huffman tree is built to send text "I am a first year student of IIT Patna." Based on the following tree what is the probable code of character 'f'?



i. 1001
ii. 1100
iii. 01111
iv. 0011

(j) What does the following program print?

```
void g(int A[],int n)
{
  int i;
  for (i = 1; i < n; ++i)
  A[i] += A[i-1];
}
main()
{
  int A[5] = {2,3,4,5,6};
  g(A,4);
  printf("%d", A[4] - A[3]);
}
```

i. 1
ii. 5
iii. 4
iv. -8

2. For each of the following questions, you can assume that the necessary header files are included.
$(10 \times 3) = 30$

(a) Consider the following program-

```
int main(void)
{
   int var1 = 0;
   const int* ptr = &var1;
   *ptr = 1;
   printf("%d\n", *ptr);
   return 0;
}
```

Point out the problem in the above program if there is any otherwise print the output

Answer: _____

Justification: _____

_____

(b) Will the following program print the average of the integers in the array? Give justification.

```
float avg(int *a, int size)
{
  int i;
  float total = 0.;
  for(i = 0; i < size; i++)
  total+=(*a)+i;
  return total/size;
}
int main()
{
  int array[4] = {2, 4, 6, 8};
  printf("Average : %f",avg(array,4));
  return 0;
}
```

Answer: _____

Justification: _____

_____

(c) What the following program prints?

```
struct test1
{
   int x1;
```

```
    char y1;
};

union test2
{
    int x2;
    char y2;
};

int main()
{
    printf("sizeof(test1)=%d, sizeof(test2)=%d,"
          ,sizeof(test1),sizeof(test2));
    return 0;
}
```

Answer: _____

Justification:_____

_____

(d) Consider the following definition of a node. What the following *fun1* function performs and returns? Assume *head* is the header pointer of a linked list.

```
struct node
{
  int data;
  struct node *next;
};
struct node *fun1(struct node *head, int value)
{
  struct node *p;
  p = head;
  while (p!=NULL)
  {
    if (p->data == value)
    return p;
    p = p->next;
  }
  return p;
}
```

Answer: _____

Justification:_____

_____

(e) Consider the following definition of a node. What the following *fun2* function performs and returns? Assume *head* is the head pointer of a linked list.

```
struct node
{
  int data;
  struct node *next;
};
struct node *fun2(struct node *head, int value)
{
  struct node *p;
  p = (struct node *) malloc(sizeof(struct node));
  p->data = value;
  p ->next = head;
  return p;
}
```

Answer: _____

Justification:_____

_____

(f) For stack implementation using linked list the following functions are given. Please check whether *push* and *pop* functions are correctly specified. If there is any error, mention it and correct it.

```
struct element
{
    char data;
struct element *next;
};
typedef struct element node;
node *top;

void initialize()
{
    top = NULL;
}

void push(char d)
{
     node *tmp;
     tmp=malloc(sizeof(node));
     tmp->data = d;
     tmp->next=top;
     top=tmp;
}

char pop()
{
     node *tmp; char d;
     tmp=top;
     top = top->next;
     d = tmp -> data;
     free(tmp);
 return d;
}
```

Answer: _____

Justification:_____

_____

(g) What the following program prints? Assume that there exists a file *Test.txt* with contents "This is a CS102 endsem exam"

```
int main()
{
  FILE *fp;
  char buf[50];
  fp = fopen("Test.txt","r");
  fseek(fp,10,SEEK_SET);
  fscanf(fp,"%s",buf);
  printf("\n%s",buf);
  fclose(fp);
```

```
  return 0;
}
```

Answer: _____

_____

Justification:_____

_____

(h) Construct the max heap by inserting the elements in following order: 3 ,2, 1, 15, 5, 4, 45. Show each insertion step.
Answer:

(i) Delete the root node from the heap tree that you have constructed in previous question. Show the tree after deletion. You must show the intermediate steps. Answer:

(j) Insert the following elements in an AVL tree. The elements are 10, 20, 30, 25, 26, 27. Show each steps.
Answer:

3. (a) We are given a linked list and we want to traverse it in a way such that the links are reversed by the time we complete the traversal. Complete the function traverseReverse that achieves this objective. You can assume that the input pointer is not NULL . Do not use any new variables or malloc calloc instructions your program should accomplish the goal by careful manipulation of the pointers. 10 Marks

```
struct node {
char data;
struct node *link;
};

struct node *traverseReverse ( struct node *p1 )
{
struct node *p2, *p3;
/* p2 points the part of the list that has already been
reversed and p1 points to the part that is yet to be
reversed. p3 may be used as an auxiliary pointer. */
/* Initialize the pointers */
p2 = p1; p1 = p1->link; p2->link = NULL;
/* As long as the unprocessed part contains more
nodes to process */

while ( _____ ) {
/* Reverse another link from the unprocessed part.
Use pointer assignments only. */
_____
_____
_____
_____

}
p1->link = p2; /* Link p1 to head the reversed list
*/
return (_____); /* Return a pointer to the header of
the reversed list */
}
```

(b) For a pair of real numbers a, b with a < b, the interval [a, b] is defined as the set of all real numbers between a and b, i.e., [a, b] = {x: x >= a and x <= b}. Represent an interval by the following structure:
typedef struct { float a, b; } interval;
We are given an array A of n intervals [a_i, b_i] for i = 0, 1, . . . , n - 1. We look at the union of all these n - 1 intervals, namely, at the set $S=\cup_{i=1}^{n-1}[a_i, b_i]$ The given intervals may be overlapping. Our task is to write S as the union of non-overlapping intervals, i.e., as an array B of m non-overlapping intervals [c_j , d_j] for j = 0, 1, . . . , m - 1 (for some m<= n). As an example, consider the ten intervals: [3, 5], [8, 12], [10, 19], [11, 18], [19, 28], [23, 29], [27, 31], [33, 42], [33, 40], [35, 38]. The corresponding array of non-overlapping intervals is as follows. We have m = 3 in this case. [3, 5], [8, 31], [33, 42]. In order to solve this problem, we sort the input intervals with respect to their left end-points. Then we enter a loop. We store the next unprocessed interval [a_i, b_i] in a temporary structure variable t . Then we look at the interval [a_{i+1}, b_{i+1}]. If a_{i+1} <= t.b , the interval [a_{i+1}, b_{i+1}] can be merged with t . This merging step alters the right end-point of t if and only if the right end-point of [a_{i+1}, b_{i+1}] is strictly bigger than that of b. Then consider the intervals [a_{i+2}, b_{i+2}] , [a_{i+3}, b_{i+3}] ,..., [a_{i+k}, b_{i+k}] as long as merging can be done. When merging is no longer possible, we write t in the output array B. We then go to the top of the loop, store [a_{i+k+1}, b_{i+k+1}] in t, and repeat the same procedure, until all of the n input intervals are taken care of. Complete the following function that implements this algorithm. The function should return the number of intervals in the output array. You do not have to write the sorting function sortIntervals() . 10 Marks

```
int mergeIntervals ( interval A[] , int n , interval B[] )
/* A is the input array of size n and B is the output
array*/
{
int i = 0, j = 0; /* i is for reading from A, j is for
writing to B*/
interval t; /* temporary variable */
sortIntervals(A,n); /* Sort A with respect to the left
end-points*/
while (1) { /* Store in t the next unprocessed input
interval*/
t = _____;
++i;
/* As long as merging is possible*/
while ((i < n) && (_____)) {
/* Look at the possibility of extending the right
end-point of t*/
if (_____)
t.b =_____;
++i;
}
/* Write the accumulated interval t to the output
array B*/
_____
_____
/* Check whether all input intervals are processed*/
if (_____) return _____; /* Return the size of B */
}
}
```

4. Use the following definition of a node of a singly linked list and write the following two functions.

```
struct node
{
  int data;
  struct node *next;
};
 typedef struct node *NODEPTR;
```

10x2=20 Marks

(a) A function which takes a head node of a linked list of integers, and removes all the odd integers from the list. It returns the head pointer after removing all the odd integers from the list.

Answer: `NODEPTR removeOdd (NODEPTR head)`

(b) The function takes a head node of a linked list of integers, an integer i and another integer d. It inserts d after the i th element in the list. If $i = 0$, then it enters at the beginning of the list.

Answer: `NODEPTR insert (NODEPTR L, int i, int d)`