**Screening Test**

NAME: Roll No.:

Mobile: Email:

**Time: 90 Minutes Max. marks: 150**

Instructions: (a) Mark you answer in the question paper and submit the paper at the end

(b) All questions are compulsory (c) No Negative marking

**Part A**

Each question carries 4 Marks (20\*4)

1. A machine needs a minimum of 100 sec to sort 1000 names by merge sort. The minimum time needed to sort 100 names will be approximately ?

(a) 50.2 sec (b) 6.7 sec (c) 72.7 sec (d) 11.2 sec

2. The postfix expression for \* + a b - c d is?

(a) ab + cd - \* (b) ab cd + - \* (c) ab + cd \* - (d) ab + - cd \*

|  |
| --- |
| 3. A list of n strings, each of length n, is sorted into lexicographic order using the merge-sort algorithm.  The running time of this computation is  (a) O(n log n) (b) O(n2 log n) (c) O(n2 + logn) (d) O(n2) |

|  |
| --- |
| 4. Given a binary tree whose inorder and preorder traversal are given by  Inorder  : EICFBGDJHK  Preorder : BCEIFDGHJK  The post order traversal of the above binary tree is  (a) IEFCGJKHDB (b) IEFCJGKHDB (c) IEFCGKJHDB (d) |

|  |
| --- |
| 5. If h is any hashing function and is used to hash n keys in to a table of size m, where n<=m, the expected  number of collisions involving a particular key x is  (a) Less than 1 (b) less than n (c) Less than m (d) Less than n/2 |

6. What does the following code do to the parameter n?

int Mystery(Tree\* tree, int &n) //Tree is a structure

{

if(tree != NULL) {

n++;

Mystery(tree->left, n);

Mystery(tree->right, n);

}

}

A. Returns the height of the tree B. Returns the number of leaves in the tree

C. Returns the number of nodes in the tree D. Returns (total nodes -1)

7. A *k-*ary tree is a tree in which every node has at most *k* children. In a *k*-ary tree with *n* nodes and height *h*, which of the following is an upper bound for the maximum number of leaves as a function of *h*, *k*, and *n?* (A) log*k n* (B) log*k h*  (C) *k h* (D) *hk*

8. Mergesort works by splitting a list of *n* numbers in half, sorting each half recursively, and merging the two halves. Which of the following data structures will allow mergesort to work in *O*(*n* log *n*) time?

I. A singly linked list

II. A doubly linked list

III. An array

1. None (B) III only (C) I and II only (D) II and III only (E) I, II, and III

9. The relation book(title,price) contains the titles and prices of different books. Assuming that no 2 books have the same process, what does the following SQL query will give:

Select title

From book as B

Where (select count(\*) from book as T where T.price>B.price)<5

(a) Titles of the 4 most expensive books (b) Titles of the 4th most expensive books

(c) Titles of the 5 most expensive books (d) Titles of the 5th expensive books

(e) Titles of all the book less than a particular price

10. Given the function definition

void calc (int a, int& b)

{ int c;

c = a + 2; a = a \* 3; b = c + a;}

What is the output of the following code fragment that invokes calc?

(All variables are of type *int*)

x = 1; y = 2; z = 3;

calc(x, y);

cout << x << ' ' << y << ' ' << z << endl;

(a) 1 2 3 (b) 1 6 3 (c) 3 6 3 (d) 1 14 9 (e) None of these

11. Study the following program

void mixtra (int x, int \*y)

{

int c;

c = x + \*y;

x = c + 1;

\*y = c + 1;

c = c + 1;

cout<< x << \*y << c;

}

int main()

{

int a, b;

int \*c;

a = 1;

b = 2;

c = &b;

/\*\*\*/

cout<< a <<b<< \*c;

}

At /\*\*\*/ a call to mixtra is done. For which of the following call, the output is: 333322

(a) mixtra(a,&b); (b) mixtra(b,&a); (c) mixtra(a,&a); (d) mixtra(a,c) (e) mixtra(b,c);

12. What is the output of the following program?  
int fun(int, float);  
int x = 6;

int main() {

int x = 4;

cout << (x + ::x + fun(::x, (float) x) );

cin >> x;

return 0;}

int fun(int x, float y) {return x - y + ::x;}

(a) 8 (b) 10 (c) 18 (d) 12 (e) 14

13. What is the output of the following program?

int rek(int x);

int main(void) {

int tal=5;

while (tal < 40) {

tal = rek(tal);

cout<< tal<< ” \t”; }

return 0; }

int rek(int x) {

if (x < 0) return -x;

else return x+rek(x-2);

}

(a) 10 32 274 (b) 10 32 152 (c) 8 32 152 (d) 10 18 274 (e) 8 18 152

1. Consider the following C code:

if (!fork()) { /\* child \*/

printf("Child ");

execlp("sleep","sleep","1",NULL); /\* execute sleep 1 => sleep 1 sec. \*/

printf(“Hello”);

exit(1); /\* exit on error \*/

}

else { /\* parent \*/

wait(NULL); /\* wait for child to terminate \*/

printf("Parent ");

}

exit(0);

What is the possible output, if none of the function calls fail?

* 1. Parent (b) Child Hello Parent (c) Parent Child Hello (d) Child Parent.

1. None of the above
2. Consider a doubly linked list. If you want to delete an element pointed to by the curr (assume not at end), which of the following codes can accomplish this:
3. curr 🡪 prev = curr 🡪 next;

curr 🡪 next = curr 🡪 prev;

1. curr 🡪 prev = curr 🡪 next 🡪 prev;

curr 🡪 next = curr 🡪 prev 🡪 next;

1. curr 🡪 prev = curr 🡪 next 🡪 next;

curr 🡪 next = curr 🡪 prev 🡪 prev;

1. curr 🡪 next 🡪 prev = curr 🡪 prev;

curr 🡪 prev 🡪 next = curr 🡪 next;

1. curr 🡪 next 🡪 next = curr 🡪 prev;

curr 🡪 prev 🡪 prev = curr 🡪 next;

1. Give asymptotic upper bound for T(n). Assume that T(n) is constant for sufficiently small n. Make your bound as tight as possible. T(n) = T(n - 2) + 2 log n

(a) O(log n) (b) O(n) (c) O(n3/2) (d) O(n log n) (e) O(n2)

17. What is the output of the following program:

void main( ){

int A[3] = {33, 21, 34};

for (int i = 0; i < 3; i++) {

for (int j = i+1; j < 3; j++){

if (A[j] > A[i]){

A[i] += A[j]; A[j] = A[i] - A[j];

A[i] = A[i] - A[j];

}

}

}

for (i=0; i<3; i++) cout << A[i] << " " ;}

(a) 21 33 34 (b) 34 33 21 (c) 33 21 34 (d) 34 21 33

18. What is the output of the following program?

int main()

{ float f =5.4; int x=2;

float result = (f/x)\*3 + 2;

int result1 = (f/x)\*3 + 2;

float result2 = (int)(f/x)\*3/4.0 + 2;

printf("%f %d %f",result,result1,result2);}

1. 10.1 10 3.5 (b) 10.1 10 3 (c) 10.1 8 3.5 (d) 10.1 8 3

19. The following is a recursive function. Assuming the input parameter *n* is a non-negative integer, what is the returned value for rec(3) and rec(4) ?

int rec(const int n)

{ if (n < 2) return n;

return rec(n-1) + rec(n-2)+1 }

(A) 3, 6 (B) 4,7 (C) 7,12 (D) 6,11

20. What is the output of the following program?

void operation(const int a[], const int b[], int c[], int size) {

for (int i=0; i<size; i++)

c[size-1-i] = abs(a[i]-b[i]); }

void main() {

int A[4] = {1,2,3,4}; int B[4] = {-6,-3,0,3}; int C[4];

operation(A,B,C,4);

for (int i=0; i<4; i++)

cout << C[i] << " ";

}

(a) 7 5 3 1 (b) 1 3 5 7 (c) 7 3 1 5 (d) 5 1 3 7

**Part B**

QB1. Fill the table below. The first row is filled as a sample. (10\*4 Marks)

|  |  |  |
| --- | --- | --- |
| Problem  (Assume the array has n elements in each case) | Order of  Brute Force Algorithm | Order of Best algorithm that you can think |
| Sorting an Array  (Note: Counting Sort is not a general algorithm) | O(n2) | O(nlogn) |
| 1. Algorithm that tells whether an array is sorted? |  |  |
| 2. Algorithm that finds the hundredth minimum element of an array? |  |  |
| 3. Algorithm that finds the intersection of two arrays each with n elements? |  |  |
| 4. Algorithm that tells whether all element of an array are the same? |  |  |
| 5. Find the median (middle) value in an unsorted array |  |  |
| 6. Inserting an element in a sorted array so that it stays sorted |  |  |
| 7. To find heavy hitter in an unsorted array. A number x is a heavy hitter in A if x occurs in A at least n/2 times. |  |  |
| 8. Algorithm to find largest substring that is a palindrome in a string of n character? |  |  |
| 9. Algorithm to find duplicates in an array? |  |  |
| 10. Algorithm to find an integer square root x of an element n where x is largest number such that x\*x <= n |  |  |

QB2. What is the running time of each of the following algorithm in theta notation? (10\*3 Marks)

(a) To construct a heap from n elements given in an array

(b) To find 10th largest element from a max heap of n elements

(c) To find the middle of a queue with n elements

(d) To insert a key in a balanced BST with n nodes

(e) To perform a DFS on a graph with n vertices and m edges

(f) To reverse a linked list with n nodes

(g) To evaluate a postfix expression with n operations

(h) To insert a node with key = k in a sorted linked list with n nodes

(i) To insert an element in a priority queue

(j) To find all prime numbers between n and 2n