**10 MCQ (1 mark each)**

Q.1. Which of the following function is used to show reverse of string?

A. reverse()

B. strrev()

C. strreverse()

D. none of the above

Q.2. Which of the following represent null character in C language?

A. 0

B. /0

C. \0

D. none of the above

Q.3. What will be the output of given code?

#include<stdio.h>

#include<string.h>

main()

{

int i;

i = strcmp(“Hi", “hello");

printf("%d",i);

}

A. error

B. -1

C. 0

D. 1

Q.4. Is the NULL pointer same as an uninitialized pointer?

A) Yes

B) No

Q.5. What will happen if in a C program you assign a value to an array element whose subscript exceeds the size of array?

A) The element will be set to 0.

B) The compiler would report an error.

C) The program may crash if some important data gets overwritten.

D) The array size would appropriately grow.

Q.6. What does the following declaration mean?

int (\*ptr)[10];

A) ptr is array of pointers to 10 integers

B) ptr is a pointer to an array of 10 integers

C) ptr is an array of 10 integers

D) ptr is an pointer to array

Q.7. Point out the error in the program

f (int a, int b)

{

int a;

a = 20;

return a;

}

A) Missing parenthesis in return statement

B) The function should be defined as int f(int a, int b)

C) Redeclaration of a

D) None of above

Q.8. How will you print \n on the screen?

A) printf("\n");

B) echo "\\n";

C) printf('\n');

D) printf("\\n");

Q.9. Which of the following statements are correct?

1: A string is a collection of characters terminated by '\0'.

2: The format specifier %s is used to print a string.

3: The length of the string can be obtained by strlen().

4: The pointer CANNOT work on string.

A) 1, 2

B) 1, 2, 3

C) 2, 4

D) 3, 4

Q.10. Which of the following statement is correct?

A) strcmp(s1, s2) returns a number less than 0 if s1>s2

B) strcmp(s1, s2) returns a number greater than 0 if s1<s2

C) strcmp(s1, s2) returns 0 if s1==s2

D) strcmp(s1, s2) returns 1 if s1==s2

**5 MCQ (2 mark each)**

Q.1. What will be the output of the following code snippet?

#include <stdio.h>

void solve() {

int first = 10, second = 20;

int third = first + second;

{

int third = second - first;

printf("%d ", third);

}

printf("%d", third);

}

int main() {

solve();

return 0;

}

A) 10 30

B) 30 10

C) 10 20

D) 20 10

Q.2. Point out the error in the program

#include<stdio.h>

int main()

{

int a[] = {10, 20, 30, 40, 50};

int j;

for(j=0; j<5; j++)

{

printf("%d\n", a);

a++;

}

return 0;

}

A) Error: Declaration syntax

B) Error: Expression syntax

C) Error: LValue required

D) Error: Rvalue required

Q.3. Point out the error in the program

#include<stdio.h>

int main()

{

int a=10;

void f();

a = f();

printf("%d\n", a);

return 0;

}

void f()

{

printf("Hi");

}

A) Error: Not allowed assignment

B) Error: Doesn't print anything

C) No error

D) None of above

Q.4. Which of the following function is correct that finds the length of a string?

A) int xstrlen(char \*s)

{

int length=0;

while(\*s!='\0')

{ length++; s++; }

return (length);

}

B) int xstrlen(char s)

{

int length=0;

while(\*s!='\0')

length++; s++;

return (length);

}

C) int xstrlen(char \*s)

{

int length=0;

while(\*s!='\0')

length++;

return (length);

}

D) int xstrlen(char \*s)

{

int length=0;

while(\*s!='\0')

s++;

return (length);

}

Q.5. What will be the output of the program ?

#include<stdio.h>

#include<string.h>

int main()

{

char str1[20] = "Hello", str2[20] = " World";

printf("%s\n", strcpy(str2, strcat(str1, str2)));

return 0;

}

A) Hello

B) World

C) Hello World

D) WorldHello

**2 Coding Questions (5 mark each)**

Q.1. String sorting

Given a string, the task is to sort the string in alphabetical order and display it as output.

**Sample Input 1**

face

**Sample Output 1**

acef

**Sample Input 2**

focus

**Sample Output 2**

cfosu

**Input Explanation**

Input consists of character value

**Output Explanation**

Output consists of sorted character value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | Complete | return | hello | Welcome | Breakfast |
| **Output** | Ceelmopt | enrrtu | ehllo | Wceelmo | aabefkrst |

**#Solution**

#include <stdio.h>

#include <string.h>

int main()

{

char string[100];

scanf("%s", string);

char temp;

int i, j;

int n = strlen(string);

for (i = 0; i < n - 1; i++)

{

for (j = i + 1; j < n; j++)

{

if (string[i] > string[j])

{

temp = string[i];

string[i] = string[j];

string[j] = temp;

}

}

}

printf("%s", string);

return 0;

}

Q.2. Print only alphabet

A string is obtained as input from the user and all the characters other than the alphabets are removed from the string and the output string containing only the alphabets is displayed.

**Sample Input 1**

We23lc333om@#e

**Sample Output 1**

Welcome

**Sample Input 2**

h@#el#$lo

**Sample Output 2**

hello

**Input Explanation**

Input consists of string value

**Output Explanation**

Output consists of string value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | my@anatomy | Chi123\*()tkara | Wel32439(\*come | Univer90\*()sity | Hor23)\*s23e |
| **Output** | Myanatomy | Chitkara | Welcome | University | Horse |

**#Solution**

#include<stdio.h>

int main()

{

char input[150];

int i, j;

gets(input);

for (i = 0; input[i] != '\0'; ++i)

{

while (!((input[i] >= 'a' && input[i] <= 'z') || (input[i] >= 'A' && input[i] <= 'Z') || input[i] == '\0'))

{

for (j = i; input[j] != '\0'; ++j)

{

input[j] = input[j + 1];

}

input[j] = '\0';

}

}

puts(input);

return 0;

}

**1 Coding Question (10 mark)**

Q.1. Number palindrome pattern

To print palindrome pyramid pattern using numbers is discussed here. Given a number n, the task is to print a palindrome pyramid containing n number of rows.

**Sample Input 1**

5

**Sample Output 1**

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

1 2 3 4 5 4 3 2 1

**Sample Input 2**

3

**Sample Output 2**

1

121

12321

**Input Explanation**

Input consists of single integer value

**Output Explanation**

Output consists of number palindrome pattern depending on the input

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 3 | 4 | 6 | 2 | 5 |
| **Output** | 1  121  12321 | 1  121  12321  1234321 | 1  121  12321  1234321  123454321  12345654321 | 1  121 | 1  121  12321  1234321  123454321 |

**#Solution**

#include<stdio.h>

int main()

{

int i, j, k, l, n;

scanf("%d", &n);

for (i = 1; i <= n; i++)

{

for (k = 1; k <= i; k++)

{

printf("%d",k);

}

for (l = i - 1; l >= 1; l--)

{

printf("%d",l);

}

printf("\n");

}

return 0;

}