**Output Questions based on String, Functions and Pointers**

Q1.

**#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;**

**}**

Explanation :

Step 1: char str1[20] = "Hello", str2[20] = " World"; The variable str1 and str2 is declared as an array of characters and initialized with value "Hello" and " World" respectively.

Step 2: printf("%s\n", strcpy(str2, strcat(str1, str2)));

=> strcat(str1, str2)) it append the string str2 to str1. The result will be stored in str1. Therefore str1 contains "Hello World".

=> strcpy(str2, "Hello World") it copies the "Hello World" to the variable str2.

**Output: "Hello World".**

Q2.

**#include < stdio.h >**

**void fun();**

**int main()**

**{ fun();**

**printf("\n");**

**return 0;**

**}**

**void fun()**

**{**

**char c;**

**if((c = getchar())!= '\n')**

**fun();**

**printf("%c", c);**

**}**

Explanation :

Step 1: void fun(); This is the prototype for the function fun().

Step 2: fun(); The function fun() is called here.

The function fun() gets a character input and the input is terminated by an enter key(New line character). It prints the given character in the reverse order.

The given input characters are "abc"

**Output: cba**

**Q3.**

**#include < stdio.h >**

**int main()**

**{**

**char str[10] = "PARAM-1000";**

**printf("%s\n", str);**

**return 0;**

**}**

Explanation :

Here str[] has declared as 10 character array and into a 10 character is stored. So there is no space for ‘\0’.This will result in overwriting of the byte beyond 10th byte reserved for '\0'.

As a result some compiler will give **Segmentation fault** and some other may print unpredictable characters after 10 characters. The process will be stopped when some where 0 is found in memory.

**Q4.**

**#include < stdio.h >**

**int main()**

**{**

**char \*names[] = { "Suresh", "Siva", "Sona", "Baiju", "Ritu"};**

**int i;**

**char \*t;**

**t = names[3];**

**names[3] = names[4];**

**names[4] = t;**

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

**printf("%s,", names[i]);**

**return 0;**

**}**

Explanation :

Step 1: char \*names[] = { "Suresh", "Siva", "Sona", "Baiju", "Ritu"}; The variable names is declared as an pointer to a array of strings.

Step 2: int i; The variable i is declared as an integer type.

Step 3: char \*t; The variable t is declared as pointer to a string.

Step 4: t = names[3]; names[3] = names[4]; names[4] = t; These statements the swaps the 4 and 5 element of the array names.

Step 5: for(i=0; i<=4; i++) printf("%s,", names[i]); These statement prints the all the value of the array names.

**Hence the output of the program is "Suresh, Siva, Sona, Ritu, Baiju".**

**Q5.**

**#include < stdio.h >**

**#include < string.h >**

**int main()**

**{**

**char str[] = "India\0\PARINAM\0";**

**printf("%d\n", strlen(str));**

**return 0;**

**}**

Explanation :

The function strlen returns the number of characters int the given string. Therefore, strlen(str) becomes strlen("India") contains 5 characters. A string is a collection of characters terminated by '\0'.

**The output of the program is 5.**

**Q6.**

**#include < stdio.h >**

**#include < string.h >**

**int main()**

**{**

**static char str1[] = "dills";**

**static char str2[20];**

**static char str3[] = "Daffo";**

**int i;**

**i = strcmp(strcat(str3, strcpy(str2, str1)), "Daffodills");**

**printf("%d\n", i);**

**return 0;**

**}**

**Explanation:**

**Step1: If we see carefully then strcpy(str2,str1) function will be executed first so “dills” will be copied in str2.**

**Step2: Then strcat(str3, and returned string by CPU that is “Dills” will be concatenated and results to str3 = “DeffoDills”**

**Step3: then the str3 will be compared with constant string “deffodills” by strcmp() function will returns zero(0) because both the string are alphabetically equal.**

**Output: 0**

**Q7.**

**#include < stdio.h >**

**#include < string.h >**

**int main()**

**{**

**static char s[] = "Hello!";**

**printf("%c\n", \*(s+strlen(s)));**

**return 0;**

}

Explanation:

Step1: **\*(s+strlen(s)) in this statement s( base address of array) + 6 strlen(s) which is length of the string “Hello!” and hence effective address is s[6]:**

**Hence output is !**

**Q8.**

**#include < stdio.h >**

**int main()**

**{**

**static char s[25] = "TheIroney man";**

**int i=0;**

**char ch;**

**ch = s[++i];**

**printf("%c", ch);**

**ch = s[i++];**

**printf("%c", ch);**

**ch = i++[s];**

**printf("%c", ch);**

**ch = ++i[s];**

**printf("%c", ch);**

**return 0;**

**}**

**Ouput: hheJ**

Step1: **ch = s[++i]; ch will be assigned with ‘h’ because i=1**

**Step2: ch = s[i++]; ch remains ‘h’ because i++ so post increment hence i=1**

Step3: **ch = i++[s]; it is equivalent to s[i++] i= 2 because of step2 post increment operation hence ch will be ’e’.**

Step4: **ch = ++i[s]; it is equivalent to ++s[i] i= 3 because of step2 post increment operation hence ch will be J because Ascii value of I will be incremented by 1.**

**Q9.**

**#include < stdio.h >**

**int main()**

**{**

**int i;**

**char a[] = "\0";**

**if(printf("%s", a))**

**printf("The string is empty\n");**

**else**

**printf("The string is not empty\n");**

**return 0;**

**}**

Explanation :

The function printf() returns the number of charecters printed on the console.

Step 1: char a[] = "\0"; The variable a is declared as an array of characters and it initialized with "\0". It denotes that the string is empty.

Step 2: if(printf("%s", a)) The printf() statement does not print anything, so it returns '0'(zero). Hence the if condition is failed.

In the else part it prints "The string is not empty".

**Q10.**

**#include < stdio.h >**

**int main()**

**{**

**char a[] = "Visual C++";**

**char \*b = "Visual C++";**

**printf("%d, %d\n", sizeof(a), sizeof(b));**

**printf("%d, %d", sizeof(\*a), sizeof(\*b));**

**return 0;**

**}**

**Answer :11, 4**

**1, 1**

Q11.

**#include < stdio.h >**

**int main()**

**{**

**char str1[] = "Hello";**

**char str2[10];**

**char \*t, \*s;**

**s = str1;**

**t = str2;**

**while(\*t=\*s)**

**\*t++ = \*s++;**

**printf("%s\n", str2);**

**return 0;**

**}**

**output: hello**

**Q12.**

**#include <stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char str[] = "India\0Super\0";**

**printf("%d\n", sizeof(str));**

**printf("%d\n", strlen(str));**

**return 0;**

**}**

**Answer :14**

**5**

**Q13.**

**#include < stdio.h >**

**int main()**

**{**

**char str[25] = "Super\0India";**

**printf("%s\n", &str+2);**

**return 0;**

**}**

Q14.

**#include < stdio.h >**

**int main()**

**{**

**char str = "India";**

**printf("%s\n", str);**

**return 0;**

**}**

**Output:Syntax error**

**Q14.**

**#include < stdio.h >**

**int main()**

**{**

**char str[] = "Nagpur";**

**str[0]='K';**

**printf("%s, ", str);**

**str = "Kanpur";**

**printf("%s", str+1);**

**return 0;**

**}**

Explanation :

The statement str = "Kanpur"; generates the LVALUE required error. We have to use strcpy function to copy a string.

To remove error we have to change this statement str = "Kanpur"; to strcpy(str, "Kanpur");

**Output: The program prints the string "anpur"**

**Q15**

**#include < stdio.h >**

**#include < string.h >**

**int main()**

**{**

**char \*str1 = "India";**

**char \*str2 = "PARAM";**

**char \*str3;**

**str3 = strcat(str1, str2);**

**printf("%s %s\n", str3, str1);**

**return 0;**

**}**

**Q16**

**#include < stdio.h >**

**int main()**

**{**

**char \*str[] = {"Frogs", "Do", "Not", "Die", "They", "Croak!"};**

**printf("%d, %d", sizeof(str), strlen(str[0]));**

**return 0;**

**}**

Explanation :

Step 1: char \*str[] = {"Frogs", "Do", "Not", "Die", "They", "Croak!"}; The variable str is declared as an pointer to the array of 6 strings.

Step 2: printf("%d, %d", sizeof(str), strlen(str[0]));

sizeof(str) denotes 6 \* 4 bytes = 24 bytes. Hence it prints '24'

strlen(str[0])); becomes strlen(Frogs)). Hence it prints '5';

**Hence the output of the program is 24, 5**

Hint: If you run the above code in Linux (32 bit platform), the output will be 24, 5 (because the size of pointer is 4 bytes).

**Q17**

**#include < stdio.h >**

**#include < string.h >**

**int main()**

**{**

**char str1[5] = "HEllo, str2[5]= "Hello";**

**int i;**

**gets(str1);**

**gets(str2);**

**i = strcmp(str1, str2);**

**printf("%d\n", i);**

**return 0;**

**}**

**Output: -32**

**Q18.**

**#include < stdio.h >**

**int main()**

**{**

**char str[10] = "India";**

**str[6] = "PARINAM";**

**printf("%s\n", str);**

**return 0;**

**}**

**Output: error**

Q19

**#include < stdio.h >**

**int main()**

**{**

**char str1[] = "Hello";**

**char str2[] = "Hello";**

**if(str1 == str2)**

**printf("Equal\n");**

**else**

**printf("Unequal\n");**

**return 0;**

**}**

**Output : unequal**

**Q20.**

Consider the following code. The function myStrcat concatenates two strings. It appends all characters of b to end of a. So the expected output is "Geeks Quiz". The program compiles fine but produces segmentation fault when run.

**#include <stdio.h>**

**void myStrcat(char \*a, char \*b)**

**{**

**int m = strlen(a);**

**int n = strlen(b);**

**int i;**

**for (i = 0; i <= n; i++)**

**a[m+i] = b[i];**

**}**

**int main()**

**{**

**char \*str1 = "Geeks ";**

**char \*str2 = "Quiz";**

**myStrcat(str1, str2);**

**printf("%s ", str1);**

**return 0;**

**}**

**Suggest the changes so that program will print "Geeks Quiz"?**