Test

Engineering Report 95000xxxx

Draft 195000xxxx Draft 1Draft 1

Date: 2018-06-28

Copyright © 2018

All Rights Reserved

**PROPRIETARY NOTICE AND DISCLAIMER**

Unless noted otherwise, this document and the information herein disclosed are proprietary to MCCI Corporation, 3520 Krums Corners Road, Ithaca, New York 14850 (“MCCI”). Any person or entity to whom this document is furnished or having possession thereof, by acceptance, assumes custody thereof and agrees that the document is given in confidence and will not be copied or reproduced in whole or in part, nor used or revealed to any person in any manner except to meet the purposes for which it was delivered. Additional rights and obligations regarding this document and its contents may be defined by a separate written agreement with MCCI, and if so, such separate written agreement shall be controlling.

The information in this document is subject to change without notice, and should not be construed as a commitment by MCCI. Although MCCI will make every effort to inform users of substantive errors, MCCI disclaims all liability for any loss or damage resulting from the use of this manual or any software described herein, including without limitation contingent, special, or incidental liability.

MCCI, TrueCard, TrueTask, MCCI Catena, and MCCI USB DataPump are registered trademarks of MCCI Corporation.

MCCI Instant RS-232, MCCI Wombat and InstallRight Pro are trademarks of MCCI Corporation.

All other trademarks and registered trademarks are owned by the respective holders of the trademarks or registered trademarks.

Copyright © 2017 by MCCI Corporation.

Document Release History

|  |  |  |
| --- | --- | --- |
| Draft 1 | 2017-MM-DD | First draft |

**Table of Contents**

1 Introduction 3

1.1 Relevant Documents 3

**List of Tables**

Table 1. References, IP Blocks 3

**List of Figures**

**List of SeQUENCE Diagrams**

## Program1:

**1.What will be the output of following program?**

#include "stdio.h"

int main()

{

    char a[] = { 'A', 'B', 'C', 'D' };

    char\* ppp = &a[0];

    \*ppp++; // Line 1

    printf("%c %c ", \*++ppp, --\*ppp); // Line 2

}

Ans:CA

Ppp points to the index 1.The printf statement is executed from right to left.—ppp decrements the value at index1.And \*++ppp points to the next location C,prints it C A

## Program2:

main()

{

if(1,0)

{

printf("True");

}

Else

{

printf("False");

}

}

Ans: false

If statement evaluates the value at the right side.if(0) so else part is executed and false is printed.

## Program3:

#### #include<stdio.h>

#### int main()

#### {

#### int a = 320;

#### char \*ptr;

#### ptr =( char \*)&a;

#### printf("%d ",\*ptr);

#### return 0;

}

Ans :64

Char pointer keep the address one byte at a time.binary value of 320 is 00000001 01000000.so pointer is pointing to first 8 bit and the decimal value is 64.

## Program4:

int f(int x, int \*py, int \*\*ppz)

{

  int y, z;

  \*\*ppz += 1;

   z  = \*\*ppz;

  \*py += 2;

   y = \*py;

   x += 3;

   return x + y + z;

}

 void main()

{

   int c, \*b, \*\*a;

   c = 4;

   b = &c;

   a = &b;

   printf( "%d", f(c,b,a));

   getchar();

}

Ans: z is changed to 5(\*\*ppz+=1). changes c to 7(\*py+=2), x is not changed. y is changed to 7. x is incremented by 3. return 7 + 7 + 5.

## Program5:

#include<stdio.h>

int main()

{ union a

{

int i;

char ch[2];

};

union a u;

u.ch[0]=3;

u.ch[1]=2;

printf("%d, %d, %d\n", u.ch[0], u.ch[1], u.i);

return 0;

}

Ans: 3,2,515

It prints the value of u.ch[0] = 3, u.ch[1] = 2 and it prints the value of u.i means the value of entire union size.

## Program 6:

7. What is the output?

#include <stdio.h>

#include <string.h>

int main(void)

{

    char\* p = "geeks";

    printf("%lu %lu %lu ", sizeof(p), sizeof(\*p), sizeof("geeks"));

    printf("%lu %lu", strlen(p), strlen("geeks"));

    return 0;

}

Ans:8 1 6 5 5

p is a pointer so sizeof returns [sizeof](http://www.geeksforgeeks.org/sizeof-operator-c/)(char\*), \*p is of type char, sizeof(“geeks”) returns the number of characters including the null character, strlen(p) returns length of string pointed by p without null character and strlen returns number of characters without null character.

## Program 7:

Int main()

{int a=5,\*b,c;

b=&a;

printf(“%d”,a\* \*b \* a+ \*b);

return 0;

}

Ans:130

25\*5+5

## Program 8:

int main()

{

    int i, j = 3;

    float k = 7;

    i = k % j;

    printf("%d", i);

    return (0);

}

Ans:error . % cannot be applied to floating point integer.

## Program 9:

int main()

{

    int x = 5;

    int \* const ptr = &x;

    ++(\*ptr);

    printf("%d", x);

    getchar();

    return 0;

}

Ans:6

++(\*ptr )increments the value of x.hence,6

## Program 10:

void fun(char\*\* str\_ref)

{

    str\_ref++;

}

int main()

{

    char \*str = (void \*)malloc(100\*sizeof(char));

    strcpy(str, "GeeksQuiz");

    fun(&str);

    puts(str);

    free(str);

    return 0;

}

Ans:GeeksQuiz

str\_ref is a local variable to fun(). When we do str\_ref++, it only changes the local variable str\_ref..

## Program 11:

int main()

{

    int x = 2;

    switch (x) {

        x--;

        switch (x) {

        case 1:

            printf("Hello");

            break;

        case 2:

            printf("GFG");

            break;

        case 3:

            printf("Welcome");

            break;

        default:

            printf("BYE");

        }

    }

    return (0);

}

Ans:No output. Since there is no outer switch case.no output.

## Program 12:

int main()

{

    int x = 2;

    switch (x) {

    case 1:

        printf("Hello");

        break;

    case 2:

        printf("Hello");

        break;

    case 3:

        printf("GFG");

        break;

    default:

        printf("BYE");

    }

    return (0);

}

Ans:Error.Since the switch case for both the cases are same,it shows error.

## Program 13:

int main()

{

    static int i=5;

    if(--i){

        main();

        printf("%d ",i);

    }

}

Ans: 0 0 0 0

Since i is a static variable and is stored in Data Section, all calls to main share same i.

## Program 14:

int fun()

{

  static int num = 16;

  return num--;

}

int main()

{

  for(fun(); fun(); fun())

    printf("%d ", fun());

  return 0;

}

Ans:14,11,8,5,2

Initially num value is 16. When initialization num value becomes 15.in the condition part num value is checked for non zero value and it is decremented to 14.When printf is called further decremented to 13 after printing 14.Likewise the loop continues till the num becomes zero.