****

|  |  |  |
| --- | --- | --- |
| **Module** | **Assessment Number** | **Assessment Type** |
| **6CS005–High Performance Computing** | **Internal Assignment 1** | **Assignment** |

Student Id : [NP03A180161]

Student Name : [Sujan Pyakurel]

Section : [L6CG9]

Module Leader : [Chiranjivi Khanal]

Submitted on : <10/09/2020>

Contents

[1. Bitwise01.c 3](#_Toc50635136)

[2. Control01.c 7](#_Toc50635137)

[3. Hello01.c 9](#_Toc50635138)

[4. Hello02.c 9](#_Toc50635139)

[5. Memory01.c 10](#_Toc50635140)

[6. Memory02.c 10](#_Toc50635141)

[7. Memory03.c 11](#_Toc50635142)

[8. Memory04.c 12](#_Toc50635143)

[9. Memory05.c 13](#_Toc50635144)

[10. Memory06.c 14](#_Toc50635145)

[11. Memory07.c 15](#_Toc50635146)

[12. Memory08.c 15](#_Toc50635147)

[13. Memory09.c 16](#_Toc50635148)

[14. Memory10.c 18](#_Toc50635149)

[15. Memory11.c 19](#_Toc50635150)

[16. Memory12.c 21](#_Toc50635151)

[17. Memory13.c 22](#_Toc50635152)

[18. Memory14.c 24](#_Toc50635153)

[19. Memory15.c 26](#_Toc50635154)

[20. Memory16.c 28](#_Toc50635155)

[21. Memory17.c 30](#_Toc50635156)

[22. Memory18.c 32](#_Toc50635157)

[23. Memory19.c 34](#_Toc50635158)

[24. Memory20.c 35](#_Toc50635159)

[25. Structs01.c 36](#_Toc50635160)

[26. Structs02.c 37](#_Toc50635161)

[27. Structs03.c 38](#_Toc50635162)

[28. Types01.c 40](#_Toc50635163)

[29. Types02.c 40](#_Toc50635164)

[30. Types03.c 41](#_Toc50635165)

[31. Types04.c 41](#_Toc50635166)

[32. Types05.c 42](#_Toc50635167)

[33. Types06.c 44](#_Toc50635168)

# 1. Bitwise01.c

#include <stdio.h>

void toBinary(unsigned char c, unsigned char \*result) {

unsigned char i;

unsigned char mask = 128;

for(i=0;i<8;i++){

if(mask & c) {

result[i]='1';

} else {

result[i]='0';

}

mask = mask >> 1;

}

result[8]=0;

}

int main() {

char result1[9];

char result2[9];

char result3[9];

char result4[9];

char result5[9];

char result6[9];

char result7[9];

char result8[9];

unsigned char n;

unsigned char mask = 110;

unsigned char or;

unsigned char and;

unsigned char not;

unsigned char leftOnce;

unsigned char leftTwice;

unsigned char rightOnce;

unsigned char rightTwice;

int i;

for(i=0;i<256;i++){

n = i;

or = n | mask;

and = n & mask;

not = ~n;

leftOnce = n << 1;

leftTwice = n << 2;

rightOnce = n >> 1;

rightTwice = n >> 2;

toBinary(n, result1);

toBinary(or, result2);

toBinary(and, result3);

toBinary(not, result4);

toBinary(leftOnce, result5);

toBinary(leftTwice, result6);

toBinary(rightOnce, result7);

toBinary(rightTwice, result8);

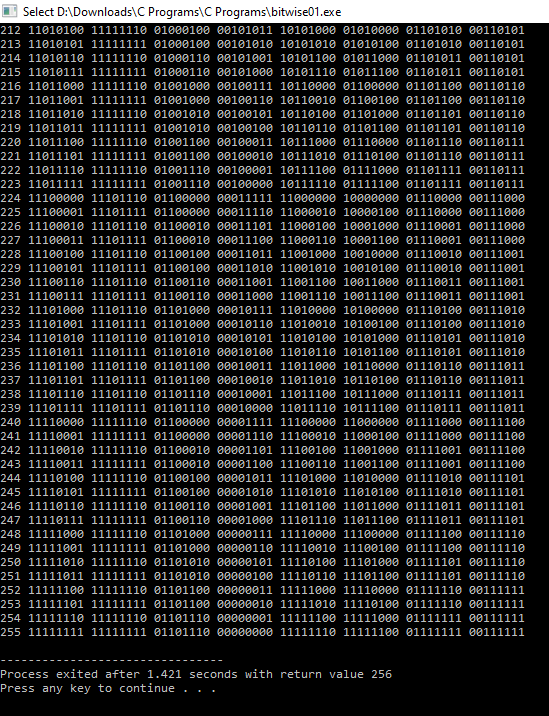
printf("%3hu %s %s %s %s %s %s %s %s\n",

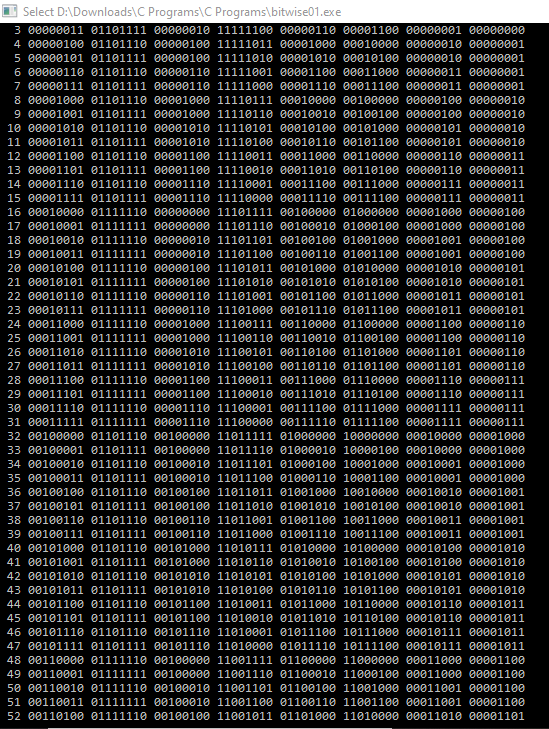
n, result1, result2, result3, result4,

result5, result6, result7, result8);

n++;

}

}



# 2. Control01.c

#include <stdlib.h>

#include <stdio.h>

int main() {

int i;

for(i=0;i<5;i++){

printf("%d,", i);

}

printf("\n");

while(i<10){

printf("%d,", i);

i++;

}

do {

printf("%d,", i);

i++;

} while(i<15);

printf("\n");

if(i>13){

printf("custard\n");

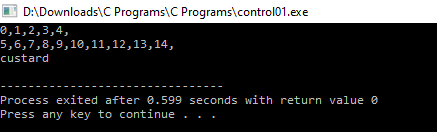
} else {

printf("gravy\n");

}

return EXIT\_SUCCESS;

}



# 3. Hello01.c

#include <stdio.h>

int main(){

printf("hello world!!\n");

return 0;

}

# Hello01.PNG

# 4. Hello02.c

#include <stdlib.h>

#include <stdio.h>

int main() {

int n = 19;

printf("Hello\nMy favourite number is %d\n", n);

return EXIT\_SUCCESS;

}

# Hello02.PNG

# 5. Memory01.c

#include <stdio.h>

#include <stdlib.h>

int main() {

int x = 123;

long int y = 321;

printf("%d %ld\n", x, sizeof(x));

printf("%ld %ld\n", y, sizeof(y));

return EXIT\_SUCCESS;

}

# Memory01.PNG

# 6. Memory02.c

#include <stdio.h>

#include <stdlib.h>

int inc(int w) {

return w + 1;

}

int main() {

int x = 123;

int y = inc(x);

printf("%d,%d\n", x, y);

return EXIT\_SUCCESS;

}

# Memory02.PNG

# 7. Memory03.c

#include <stdio.h>

#include <stdlib.h>

void inc(int \*w) {

\*w = \*w + 1;

}

int main() {

int x = 123;

int y = x;

inc(&y);

printf("%d,%d\n", x, y);

return EXIT\_SUCCESS;

}

# Memory03.PNG

# 8. Memory04.c

#include <stdio.h>

#include <stdlib.h>

void inc(int \*w) {

\*w = \*w + 1;

}

int main() {

int x = 123;

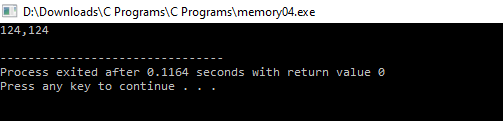
int \*y;

y = &x;

inc(y);

printf("%d,%d\n", x, \*y);

return EXIT\_SUCCESS;

}

# 9. Memory05.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

void inc(int \*w) {

\*w = \*w + 1;

}

int main() {

int \*x = malloc(sizeof(int));

\*x = 123;

int \*y;

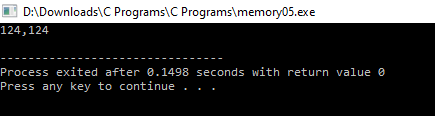
y = x;

inc(y);

printf("%d,%d\n", \*x, \*y);

free(x);

return EXIT\_SUCCESS;

}

# 10. Memory06.c

#include <stdio.h>

#include <stdlib.h>

struct pair {

int a;

int b;

};

int main() {

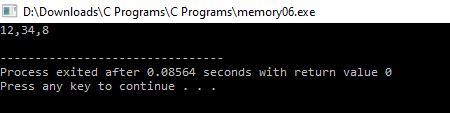
struct pair x;

x.a = 12;

x.b = 34;

printf("%d,%d,%ld\n", x.a, x.b, sizeof(struct pair));

return EXIT\_SUCCESS;

}

# 11. Memory07.c

#include <stdio.h>

#include <stdlib.h>

typedef struct {

int a;

int b;

} pair;

int main() {

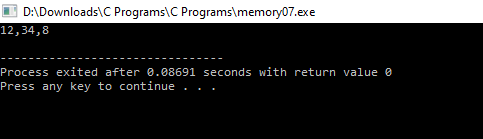
pair x;

x.a = 12;

x.b = 34;

printf("%d,%d,%ld\n", x.a, x.b, sizeof(pair));

return EXIT\_SUCCESS;

}

# 12. Memory08.c

#include <stdio.h>

#include <stdlib.h>

typedef struct {

int a;

int b;

} pair;

void inc(pair \*w) {

w->a = w->a + 1;

w->b = w->b + 1;

}

int main() {

pair x;

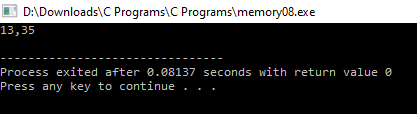
x.a = 12;

x.b = 34;

inc(&x);

printf("%d,%d\n", x.a, x.b);

return EXIT\_SUCCESS;

}

# 13. Memory09.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

typedef struct {

int a;

int b;

} pair;

void inc(pair \*w) {

w->a = w->a + 1;

w->b = w->b + 1;

}

int main() {

pair \*x;

x = malloc(sizeof(pair));

x->a = 12;

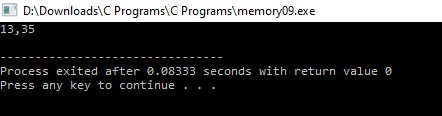
x->b = 34;

inc(x);

printf("%d,%d\n", x->a, x->b);

free(x);

return EXIT\_SUCCESS;

}

# 14. Memory10.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

int main() {

int i;

int \*x;

x = malloc(sizeof(int) \* n);

printf("%ld\n", sizeof(x));

printf("%ld\n", sizeof(\*x));

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

x[i] = 2 \* i;

}

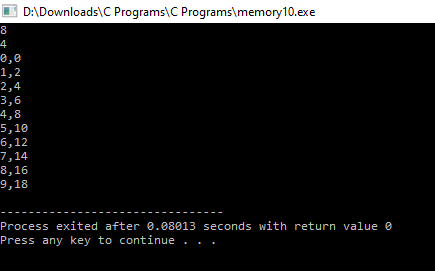
for(i=0;i<n;i++){

printf("%d,%d\n", i, x[i]);

}

free(x);

return EXIT\_SUCCESS;

}

# 15. Memory11.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

int main() {

int i;

int \*x, \*y;

x = malloc(sizeof(int) \* n);

y = x;

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

\*y = 2 \* i;

y++;

}

y = x;

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

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

y++;

}

free(x);

return EXIT\_SUCCESS;

}

# Memory11.PNG

# 16. Memory12.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

void inc(int \*w) {

int i;

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

w[i] = w[i] + 1;

}

}

int main() {

int i;

int \*x, \*y;

x = malloc(sizeof(int) \* n);

y = x;

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

\*y = 2 \* i;

y++;

}

inc(x);

y = x;

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

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

y++;

}

free(x);

return EXIT\_SUCCESS;

}

# Memory12.PNG

# 17. Memory13.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

void inc(int \*w) {

int i;

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

\*w = \*w + 1;

w++;

}

}

int main() {

int i;

int \*x, \*y;

x = malloc(sizeof(int) \* n);

y = x;

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

\*y = 2 \* i;

y++;

}

inc(x);

y = x;

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

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

y++;

}

free(x);

return EXIT\_SUCCESS;

}

# Memory13.PNG

# 18. Memory14.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int n = 10;

void initialise(int \*w) {

int i;

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

\*w = 2 \* i;

w++;

}

}

void inc(int \*w) {

int i;

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

\*w = \*w + 1;

w++;

}

}

void output(int \*w) {

int i;

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

printf("%d,%d\n", i, w[i]);

}

}

int main() {

int \*x;

x = malloc(sizeof(int) \* n);

initialise(x);

inc(x);

output(x);

free(x);

return EXIT\_SUCCESS;

}

# Memory14.PNG

# 19. Memory15.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

int w = 4;

int h = 3;

void initialise(int \*p) {

int r, c, i, j;

j=10;

for(r=0;r<h;r++){

for(c=0;c<w;c++){

i = (r\*w) + c;

p[i] = j++;

}

}

}

void inc(int \*q) {

int i;

for(i=0;i<w\*h;i++){

\*q = \*q + 1;

q++;

}

}

void output(int \*p) {

int r, c, i = 0;

for(r=0;r<h;r++){

for(c=0;c<w;c++){

printf("%d\t", p[i++]);

}

printf("\n");

}

}

int main() {

int \*p;

p = malloc(sizeof(int) \* w \* h);

initialise(p);

inc(p);

output(p);

free(p);

return EXIT\_SUCCESS;

}

# Memory15.PNG

# 20. Memory16.c

#include <stdio.h>

#include <stdlib.h>

#include <malloc.h>

typedef struct {

int a;

int b;

} pair;

int w = 4;

int h = 3;

void initialise(pair \*p) {

int r, c, i, j;

j=10;

for(r=0;r<h;r++){

for(c=0;c<w;c++){

i = (r\*w) + c;

p[i].a = j++;

p[i].b = j++;

}

}

}

void inc(pair \*q) {

int i;

for(i=0;i<w\*h;i++){

q->a = q->a + 1;

q->b = q->b + 1;

q++;

}

}

void output(pair \*p) {

int r, c, i = 0;

for(r=0;r<h;r++){

for(c=0;c<w;c++){

printf("%d,%d\t", p[i].a, p[i].b);

i++;

}

printf("\n");

}

}

int main() {

pair \*p;

p = malloc(sizeof(pair) \* w \* h);

initialise(p);

inc(p);

output(p);

free(p);

return EXIT\_SUCCESS;

}

# Memory16.PNG

# 21. Memory17.c

#include <stdio.h>

int main() {

unsigned char a = 10;

unsigned char \*b = &a;

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

unsigned char c[] = {21, 15, 92};

unsigned char \*d = c;

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

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

d++;

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

d++;

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

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

int g[] = {21, 15, 92};

int \*h = g;

printf("%p\n", h);

printf("%d ", \*h);

h++;

printf("%d ", \*h);

h++;

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

printf("%p\n", h);

unsigned int e = 197127; // (3\*65536) + (2\*256) + 7

unsigned char \*f = (char \*) &e;

printf("%ld\n", sizeof(e));

printf("%d ", \*f);

f++;

printf("%d ", \*f);

f++;

printf("%d ", \*f);

f++;

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

}

# Memory17.PNG

# 22. Memory18.c

#include <stdio.h>

void printMemory(void \*ptr, unsigned int n) {

unsigned int i;

unsigned char \*p = ptr;

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

printf("%3d ", \*p);

p++;

}

printf("\n");

}

void printMemory2(void \*ptr, unsigned int n) {

unsigned int i;

unsigned char \*p = ptr;

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

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

p++;

}

printf("\n");

}

int main(){

unsigned char a[] = {65, 66, 67, 90};

printMemory(a, 4);

printMemory2(a, 4);

int b = 197127;

printMemory(&b, 4);

char c[] = "XYZ123";

printMemory(c, 6);

double d = 65536;

printf("%lu\n", sizeof(d));

printMemory(&d, 8);

printf("%p %p %p %p\n", a, &b , c, &d);

}

# Memory18.PNG

# 23. Memory19.c

#include <stdio.h>

void f1() {

printf("cat\n");

}

void f2() {

printf("mouse\n");

}

void f3(void (\*f)() ) {

f();

}

int main() {

f3(&f1);

f3(&f2);

return 0;

}

# Memory19.PNG

# 24. Memory20.c

#include <stdio.h>

void printMemory(void \*ptr, unsigned int n) {

unsigned int i;

unsigned char \*p = ptr;

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

printf("%3d ", \*p);

p++;

}

printf("\n");

}

int add(int x) {

return x+29;

}

void doEvilThings(int (\*f)(int) ) {

char \*ptr = (char \*) f;

ptr = ptr + 12;

\*ptr = 66;

}

int main() {

printf("%d\n", add(1));

printMemory(&add, 20);

doEvilThings(&add);

printf("%d\n", add(1));

return 0;

}

# Memory20.PNG

# 25. Structs01.c

#include <stdio.h>

struct t {

unsigned int h;

unsigned int m;

unsigned int s;

};

int main() {

struct t a;

struct t \*b;

a.h = 5;

a.m = 9;

a.s = 45;

printf("Time a is %u:%02u:%02u\n", a.h, a.m, a.s);

b = &a;

printf("Time b is %u:%02u:%02u\n", b->h, b->m, b->s);

return 0;

}

# Structs01.PNG

# 26. Structs02.c

#include <stdio.h>

typedef struct {

unsigned int h;

unsigned int m;

unsigned int s;

} t;

int main() {

t a;

t \*b;

a.h = 5;

a.m = 9;

a.s = 45;

printf("Time a is %u:%02u:%02u\n", a.h, a.m, a.s);

b = &a;

printf("Time b is %u:%02u:%02u\n", b->h, b->m, b->s);

return 0;

}

# Structs02.PNG

# 27. Structs03.c

#include <stdio.h>

#include <malloc.h>

typedef struct {

unsigned int h;

unsigned int m;

unsigned int s;

} t;

int main() {

t \*a;

a = (t \*) malloc(sizeof(t));

a->h = 5;

a->m = 9;

a->s = 45;

//printf("Time a is %u:%02u:%02u\n", a->h, a->m, a->s);

printf("size is %d", sizeof(t));

//free(a);

return 0;

}

# Structs03.PNG

# 28. Types01.c

#include <stdlib.h>

#include <stdio.h>

int main() {

int x = 10;

int y = 3;

printf("%d / %d = %d\n", x, y, x/y);

return EXIT\_SUCCESS;

}

# Types01.PNG

# 29. Types02.c

#include <stdlib.h>

#include <stdio.h>

int main() {

long int x = 10L;

long int y = 3L;

printf("%ld / %ld = %ld\n", x, y, x/y);

return EXIT\_SUCCESS;

}

# Types02.PNG

# 30. Types03.c

#include <stdlib.h>

#include <stdio.h>

int main() {

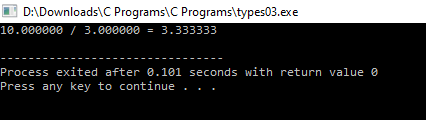
float x = 10.0f;

float y = 3.0f;

printf("%f / %f = %f\n", x, y, x/y);

return EXIT\_SUCCESS;

}



# 31. Types04.c

#include <stdlib.h>

#include <stdio.h>

int main() {

double x = 10.0;

double y = 3.0;

printf("%lf / %lf = %lf\n", x, y, x/y);

return EXIT\_SUCCESS;

}

# Types04.PNG

# 32. Types05.c

#include <stdlib.h>

#include <stdio.h>

int main() {

int a = 2;

int b = 3;

int c = 2;

int d = 4;

printf("There are no booleans in c\n");

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

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

printf("%d\n", a!=b);

printf("%d\n", a!=c);

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

printf("%d\n", !(a==b));

int e = (a == b) || (a == c);

int f = (a == b) && (a == c);

printf("e=%d\n", e);

printf("f=%d\n", f);

if(e) {

printf("e=true\n");

} else {

printf("e=false\n");

}

if(f) {

printf("f=true\n");

} else {

printf("f=false\n");

}

return EXIT\_SUCCESS;

}

# Types05.PNG

# 33. Types06.c

#include <stdlib.h>

#include <stdio.h>

int main() {

printf("Strings are just arrays of chars\n");

char \*message1 = "hello";

char \*message2 = "kevan";

printf("%s %s\n", message1, message2);

printf("Look in /usr/include/string.h for functions\n");

printf("that can be applied. Each has a man page.\n");

return EXIT\_SUCCESS;

}

# Types06.PNG