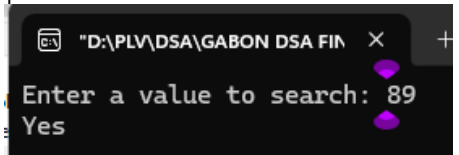
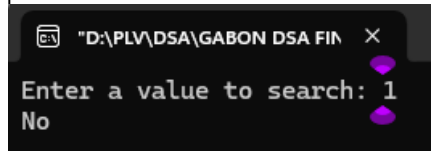


MIDTERM EXAM EXAMPLE 1

PROBLEM 1:

```
main.c X
1 #include <stdio.h>
2
3 Gabon_Johnrey_Midterm_Example1_Prog1
4 Sources
5 main.c
6 printf("Enter a value to search: ");
7 scanf("%d", &v);
8
9 for (int i = 0; i < 10; i++) {
10     if (A[i] == v) {
11         found = 1;
12         break;
13     }
14 }
15
16 if (found)
17     printf("Yes\n");
18 else
19     printf("No\n");
20
21 return 0;
22 }
```



PROBLEM 2:

```
main.c X
1 #include <stdio.h>
2
3 int main() {
4     int A[100][20];
5     for (int i = 0; i < 100; i++) {
6         for (int j = 0; j < 20; j++) {
7             A[i][j] = (i + 1) * (j + 1);
8         }
9     }
10
11     for (int i = 0; i < 100; i++) {
12         for (int j = 0; j < 20; j++) {
13             printf("%d ", A[i][j]);
14         }
15         printf("\n");
16     }
17
18     return 0;
19 }
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60
4 8 12 16 20 24 28 32 36 40 44 48 52 56 60 64 68 72 76 80
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
6 12 18 24 30 36 42 48 54 60 66 72 78 84 90 96 102 108 114 120
7 14 21 28 35 42 49 56 63 70 77 84 91 98 105 112 119 126 133 140
8 16 24 32 40 48 56 64 72 80 88 96 104 112 120 128 136 144 152 160
9 18 27 36 45 54 63 72 81 90 99 108 117 126 135 144 153 162 171 180
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
11 22 33 44 55 66 77 88 99 110 121 132 143 154 165 176 187 198 209 220
12 24 36 48 60 72 84 96 108 120 132 144 156 168 180 192 204 216 228 240
13 26 39 52 65 78 91 104 117 130 143 156 169 182 195 208 221 234 247 260
14 28 42 56 70 84 98 112 126 140 154 168 182 196 210 224 238 252 266 280
15 30 45 60 75 90 105 120 135 150 165 180 195 210 225 240 255 270 285 300
16 32 48 64 80 96 112 128 144 160 176 192 208 224 240 256 272 288 304 320
17 34 51 68 85 102 119 136 153 170 187 204 221 238 255 272 289 306 323 340
18 36 54 72 90 108 126 144 162 180 198 216 234 252 270 288 306 324 342 360
19 38 57 76 95 114 133 152 171 190 209 228 247 266 285 304 323 342 361 380
20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360 380 400
21 42 63 84 105 126 147 168 189 210 231 252 273 294 315 336 357 378 399 420
22 44 66 88 110 132 154 176 198 220 242 264 286 308 330 352 374 396 418 440
23 46 69 92 115 138 161 184 207 230 253 276 299 322 345 368 391 414 437 460
24 48 72 96 120 144 168 192 216 240 264 288 312 336 360 384 408 432 456 480
25 50 75 100 125 150 175 200 225 250 275 300 325 350 375 400 425 450 475 500
26 52 78 104 130 156 182 208 234 260 286 312 338 364 390 416 442 468 494 520
27 54 81 108 135 162 189 216 243 270 297 324 351 378 405 432 459 486 513 540
28 56 84 112 140 168 196 224 252 280 308 336 364 392 420 448 476 504 532 560
29 58 87 116 145 174 203 232 261 290 319 348 377 406 435 464 493 522 551 580
30 60 90 120 150 180 210 240 270 300 330 360 390 420 450 480 510 540 570 600
31 62 93 125 155 186 217 248 279 310 341 372 403 434 465 496 527 558 589 620
32 64 96 128 160 192 224 256 288 320 352 384 416 448 480 512 544 576 608 640
33 66 99 132 165 198 231 264 297 330 363 396 429 462 495 528 561 594 627 660
34 68 102 136 170 204 238 272 306 340 374 408 442 476 510 544 578 612 646 680
35 70 105 140 175 210 245 280 315 350 385 420 455 490 525 560 595 630 665 700
36 72 108 144 180 216 252 288 324 360 396 432 468 504 540 576 612 648 684 720
37 74 111 148 185 222 259 296 333 370 407 444 481 518 555 592 629 666 703 740
38 76 114 152 190 228 266 304 342 380 418 456 494 532 570 608 646 684 722 760
39 78 117 156 195 234 273 312 351 390 429 468 507 546 585 624 663 702 741 780
40 80 120 160 200 240 280 320 360 400 440 480 520 560 600 640 680 720 760 800
41 82 123 164 205 246 288 330 370 410 451 492 533 574 615 656 697 738 779 820
42 84 126 168 210 252 294 336 378 420 462 504 546 588 630 672 714 756 798 840
43 86 129 172 215 258 301 344 387 430 473 516 559 602 645 688 731 774 817 860
44 88 132 176 220 264 308 352 396 440 484 528 572 616 660 704 748 792 836 880
45 90 135 180 225 270 315 360 405 450 495 540 585 630 675 720 765 810 855 900
46 92 138 184 230 276 322 368 414 460 506 552 598 644 690 736 782 828 874 920
47 94 141 188 235 282 329 376 423 470 517 564 611 658 705 752 799 846 893 940
48 96 144 192 240 288 336 384 432 480 528 576 624 672 720 768 816 864 912 960
49 98 147 196 245 294 343 392 441 490 539 588 637 686 735 784 832 882 931 980
50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000
```

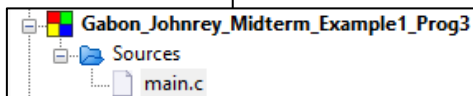
```
51 102 153 204 255 306 357 408 459 510 561 612 663 714 765 816 867 918 969 1020
52 104 156 208 260 312 364 416 468 520 572 624 676 728 780 832 884 936 988 1040
53 106 159 212 265 318 371 424 477 530 583 636 689 742 795 848 901 954 1007 1060
54 108 162 216 270 324 378 432 486 540 594 648 702 756 810 864 918 972 1026 1080
55 110 165 220 275 330 385 440 495 550 605 660 715 770 825 880 935 990 1045 1100
56 112 168 224 280 336 392 448 504 560 616 672 728 784 840 896 952 1008 1064 1120
57 114 171 228 285 342 399 456 513 570 627 684 741 798 855 912 969 1026 1083 1140
58 116 174 232 290 348 406 464 522 580 638 696 754 812 870 928 986 1044 1102 1160
59 118 177 236 295 354 413 472 531 590 649 708 767 826 885 944 1003 1062 1121 1180
60 120 180 240 300 360 420 480 540 600 660 720 780 840 900 960 1020 1080 1140 1200
61 122 183 244 305 366 427 488 549 610 671 732 793 854 915 976 1037 1098 1159 1220
62 124 186 248 310 372 434 496 558 620 682 744 806 868 930 992 1054 1116 1178 1240
63 126 189 252 315 378 441 504 567 630 693 756 819 882 945 1008 1071 1134 1197 1260
64 128 192 256 320 384 448 512 576 640 704 768 832 896 960 1024 1088 1152 1216 1280
65 130 195 260 325 390 455 520 585 650 715 780 845 910 975 1040 1105 1170 1235 1300
66 132 198 264 330 396 462 528 594 660 726 792 858 924 990 1056 1122 1188 1254 1320
67 134 201 268 335 402 469 536 603 670 737 804 871 938 1005 1072 1139 1206 1273 1340
68 136 204 272 340 408 476 544 612 680 748 816 884 952 1020 1088 1156 1224 1292 1360
69 138 207 276 345 414 483 552 621 690 759 828 897 966 1035 1104 1173 1242 1311 1380
70 140 210 280 350 420 490 560 630 700 770 840 910 980 1050 1120 1190 1260 1330 1400
71 142 213 284 355 426 497 568 639 710 781 852 923 994 1065 1136 1207 1278 1349 1420
72 144 216 288 360 432 504 576 648 720 792 864 936 1008 1080 1152 1224 1296 1368 1440
73 146 219 292 365 438 511 584 657 730 803 876 949 1022 1095 1168 1241 1314 1387 1460
74 148 222 296 370 444 518 592 666 740 814 888 962 1036 1110 1184 1258 1332 1406 1480
75 150 225 300 375 450 525 600 675 750 825 900 975 1050 1125 1200 1275 1350 1425 1500
76 152 228 304 380 456 532 608 684 760 836 912 988 1064 1140 1216 1292 1368 1444 1520
77 154 231 308 385 462 539 616 693 770 847 924 1001 1078 1155 1232 1309 1386 1463 1540
78 156 234 312 390 468 546 624 702 780 858 936 1014 1092 1170 1248 1326 1404 1482 1560
79 158 237 316 395 474 553 632 711 790 869 948 1027 1106 1185 1264 1343 1422 1501 1580
80 160 240 320 400 480 560 640 720 800 880 960 1040 1120 1200 1280 1360 1440 1520 1600
81 162 243 324 405 486 567 648 729 810 891 972 1053 1134 1215 1296 1377 1458 1539 1620
82 164 246 328 410 492 574 656 738 820 902 984 1066 1148 1230 1312 1394 1476 1558 1640
83 166 249 332 415 498 581 664 747 830 913 996 1079 1162 1245 1328 1411 1494 1577 1660
84 168 252 336 420 504 588 672 756 840 924 1008 1092 1176 1260 1344 1428 1512 1596 1680
85 170 255 340 425 510 595 680 765 850 935 1020 1105 1190 1275 1360 1445 1530 1615 1700
86 172 258 344 430 516 602 688 774 860 946 1032 1118 1204 1290 1376 1462 1548 1634 1720
87 174 261 348 435 522 609 696 783 870 957 1044 1131 1218 1305 1392 1479 1566 1653 1740
88 176 264 352 440 528 616 704 792 880 968 1056 1144 1232 1320 1408 1496 1584 1672 1760
89 178 267 356 445 534 623 712 801 890 979 1068 1157 1246 1335 1424 1513 1602 1691 1780
90 180 270 360 450 540 630 720 810 900 990 1080 1170 1260 1350 1440 1530 1620 1710 1800
91 182 273 364 455 546 637 728 819 910 1001 1092 1183 1274 1365 1456 1547 1638 1729 1820
92 184 276 368 460 552 644 736 828 920 1012 1104 1196 1288 1380 1472 1564 1656 1748 1840
93 186 279 372 465 558 651 744 837 930 1023 1116 1209 1302 1395 1488 1581 1674 1767 1860
94 188 282 376 470 564 658 752 846 940 1034 1128 1222 1316 1410 1504 1598 1692 1786 1880
95 190 285 380 475 570 665 760 855 950 1045 1140 1235 1330 1425 1520 1615 1710 1805 1900
96 192 288 384 480 576 672 768 864 960 1056 1152 1248 1344 1440 1536 1632 1728 1824 1920
97 194 291 388 485 582 679 776 873 970 1067 1164 1261 1358 1455 1552 1649 1746 1843 1940
98 196 294 392 490 588 686 784 882 980 1078 1176 1274 1372 1470 1568 1666 1764 1862 1960
99 198 297 396 495 594 693 792 891 990 1089 1188 1287 1386 1485 1584 1683 1782 1881 1980
100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000
```

PROBLEM 3:

```
#include <stdio.h>
#include <string.h>

struct House {
    int id;
    int NumberOfRooms;
    char address[50];
    int OwnerPhone;
};

int main() {
    struct House A[100]; // Array of 100 houses
    printf("Struct House created sucessfully\n");
    return 0;
}
```



Struct House created sucessfully

PROBLEM 4:

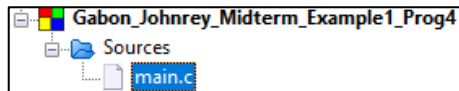
```
#include <stdio.h>
#include <string.h>

struct House {
    int id;
    int NumberOfRooms;
    char address[100];
    int OwnerPhone;
};

int main() {
    struct House A[100];
    A[0].id = 0;
    A[0].NumberOfRooms = 5;
    strcpy(A[0].address, "Shubra Street on the front of the old hospital, Building 5, Floor 4");
    A[0].OwnerPhone = 555779922;

    // Print the details of the house
    printf("House ID: %d\n", A[0].id);
    printf("Number of Rooms: %d\n", A[0].NumberOfRooms);
    printf("Address: %s\n", A[0].address);
    printf("Owner Phone: %d\n", A[0].OwnerPhone);

    return 0;
}
```



House ID: 0
Number of Rooms: 5
Address: Shubra Street on the front of the old hospital, Building 5, Floor 4
Owner Phone: 555779922

PROBLEM 5:

```
#include <stdio.h>

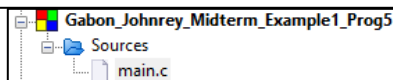
struct House {
    int id;
    int NumberOfRooms;
    char address[50];
    int OwnerPhone;
};

int main() {
    struct House A[100];

    // Initialize the NumberOfRooms for each house
    for (int i = 0; i < 100; i++) {
        A[i].NumberOfRooms = (i + 1) % 10;
    }

    // Display the number of rooms for houses with more than 5 rooms
    for (int i = 0; i < 100; i++) {
        if (A[i].NumberOfRooms > 5) {
            printf("%d\n", A[i].NumberOfRooms);
        }
    }

    return 0;
}
```

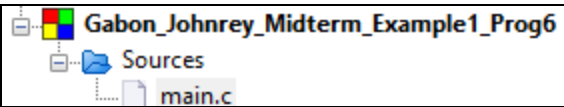


6
7
8
9
6
7
8
9
6
7
8
9
6
7
8
9
6
7
8
9
6
7
8
9

PROBLEM 6:

```
#include <stdio.h>
```

```
int main() {  
    int *ptr, a[10], x;  
    x = 10;  
    a[0] = 1;  
    a[5] = 10;  
    a[7] = 5;  
    ptr = &x;  
    printf("%d\n", x); // Outputs 10  
    *ptr = *ptr + a[5];  
    printf("%d\n", x); // Outputs 20  
  
    return 0;  
}
```

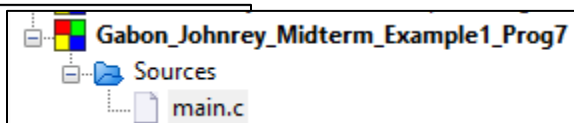


```
10  
20
```

PROBLEM 7:

```
#include <stdio.h>
```

```
int main() {  
    int *ptr, a[10];  
    ptr = &a[0];  
    a[0] = 1;  
    a[4] = 4;  
    a[5] = 4;  
    a[6] = 4;  
    a[7] = 4;  
    a[8] = -5;  
    ptr += 6;  
    ptr--;  
    *ptr = *ptr + a[5];  
    printf("%d\n", *ptr); // Outputs 8  
  
    return 0;  
}
```



```
8
```

PROBLEM 8

```
#include <stdio.h>
```

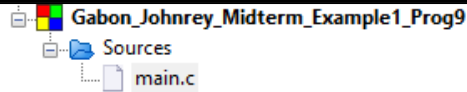
```
struct NodeStudent {  
    int id;  
    char name[20];  
    int age;  
    char address[20];  
};  
int main()  
{  
    printf("Created struct NodeStudent successfully\n");  
    return 0;  
}
```



```
Created struct NodeStudent successfully
```

PROBLEM 9:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct NodeStudent
{
    int id;
    char name[20];
    int age;
    char address[20];
    struct NodeStudent *next;
};
struct NodeStudent *head = NULL;
int SizeofTheList()
{
    struct NodeStudent *curr = head;
    int count = 0;
    while (curr != NULL)
    {
        count++;
        curr = curr->next;
    }
    return count;
}
void addNode(int id, const char *name, int age, const char *address)
{
    struct NodeStudent *newNode = (struct NodeStudent*)malloc(sizeof(struct NodeStudent));
    newNode->id = id;
    strcpy(newNode->name, name);
    newNode->age = age;
    strcpy(newNode->address, address);
    newNode->next = head;
    head = newNode;
}
int main()
{
    addNode(1, "Alice", 20, "123 Main St");
    addNode(2, "Bob", 21, "456 Elm St");
    addNode(3, "Charlie", 22, "789 Oak St");
    printf("Size of the list: %d\n", SizeofTheList());
    return 0;
}
```



Size of the list: 3

PROBLEM 10:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct NodeStudent {
    int id;
    char name[20];
    int age;
    char address[20];
    struct NodeStudent *next;
};

struct NodeStudent *head = NULL;

void DisplayNames(int minAge, int maxAge) {
    struct NodeStudent *curr = head;

    while (curr != NULL) {
        if (curr->age >= minAge && curr->age <= maxAge) {
            printf("%s\n", curr->name);
        }
        curr = curr->next;
    }
}

void addNode(int id, const char *name, int age, const char *address) {
    struct NodeStudent *newNode = (struct NodeStudent*)malloc(sizeof(struct NodeStudent));
    newNode->id = id;
    strcpy(newNode->name, name);
    newNode->age = age;
    strcpy(newNode->address, address);
    newNode->next = head;
    head = newNode;
}

int main() {
    addNode(1, "Alice", 20, "123 Main St");
    addNode(2, "Bob", 25, "456 Elm St");
    addNode(3, "Charlie", 28, "789 Oak St");
    addNode(4, "David", 30, "101 Pine St");
    addNode(5, "Eve", 22, "202 Maple St");

    printf("Names of students aged between 24 and 30:\n");
    DisplayNames(24, 30);

    return 0;
}
```

```
Names of students aged between 24 and 30:
David
Charlie
Bob
```