



Instituto Politécnico Nacional Escuela Superior de Cómputo



Paradigmas de Programación

Actividad: Práctica 1 de Laboratorio

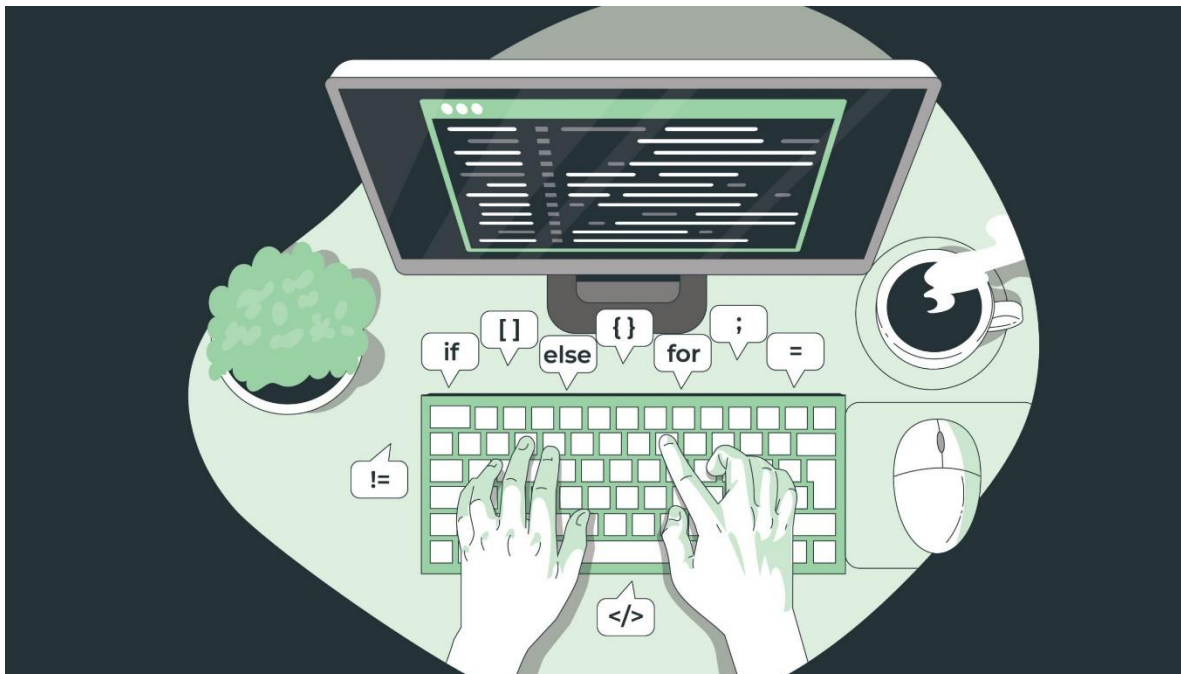
Profesor: García Floriano Andrés

Fecha: 5/03/2024

Alumno:

Pacheco Refugio Alan Ivan

Grupo: 3CV1



Reporte Práctica 1 de Laboratorio de Paradigmas de Programación

Inicialmente el problema nos plantea la creación de varias funciones las cuales son, generación, impresión, búsqueda, ejemplificación y cálculo de números aleatorios, estos números inicialmente son generados a partir del uso de la semilla `srand(NULL());` la cual nos puede ayudar a partir de la librería `#include <time.h>`, de esta manera tendríamos el primer punto del problema el cual es generar 1000 números aleatorios, una vez generados hay que imprimirlos en donde haciendo uso de `void's` y funciones `switch` pude hacer un menú que ejemplificara o ayudara de mejor manera a entender lo que queremos hacer.

De esta manera el primer paso es inicializar el número de números aleatorios que queremos en este caso son 1000

```
C:\Users\ivano\OneDrive>cd "Escritorio (3211)"

C:\Users\ivano\OneDrive\Escritorio (3211)>gcc prueba2.c

C:\Users\ivano\OneDrive\Escritorio (3211)>a
Ingrese el tamaño de 1000 números aleatorios |
```

A partir de este momento los números se generan y procedemos a imprimirlos en pantalla

```
902 245 168 7 911 760 982 367 735 316 674 614 946 856 245 369 634 598 87 361 380 241 661 174 209 404 323 787 3
202 661 411 295 377 892 793 771 310 903 285 751 903 179 49 131 445 747 695 979 378 750 822 222 320 47 666 682
21 386 403 321 714 867 843 220 908 579 941 847 858 538 730 87 725 494 612 868 13 162 206 413 193 310 806 729 7
585 483 922 633 266 122 101 180 134 648 733 468 829 665 798 702 736 190 613 612 335 808 593 85 451 538 957 26
251 245 610 648 227 142 248 410 4 20 554 143 434 412 179 610 849 891 101 330 17 663 546 768 330 648 623 464 60
8 29 606 100 101 612 721 315 251 145 995 616 943 594 604 359 799 500 761 446 154 33 365 967 621 381 686 674 93
42 533 578 381 238 874 798 475 883 125 276 130 672 185 924 956 899 606 814 9 178 94 702 324 197 830 82 456 330
90 560 565 859 576 192 959 840 514 968 480 244 608 523 357 384 389 390 377 952 722 459 862 472 462 899 596 488
98 777 414 897 371 624 129 578 163 275 568 115 268 922 78 951 603 272 902 24 726 257 917 128 361 126 778 571 8
772 380 241 238 246 169 416 671 996 648 974 141 770 176 133 882 957 651 57 807 360 238 402 73 9 312 625 513 54
560 23 408 503 698 661 677 500 227 726 228 218 47 678 397 566 990 685 116 318 961 488 768 850 294 375 323 872
0 363 303 807 273 379 771 567 352 467 283 800 72 231 648 416 323 628 602 336 735 109 762 863 606 18 684 712 12
145 901 295 441 93 120 559 48 221 154 166 257 162 700 572 848 159 386 752 309 484 828 54 925 309 78 650 204 72
114 365 891 253 798 906 981 672 129 174 243 700 725 509 751 977 397 450 754 161 398 995 139 722 239 411 17 745
96 617 826 354 238 436 127 877 8 457 463 113 818 198 190 215 523 360 825 595 562 105 856 876 460 553 506 887 2
72 790 410 748 601 62 141 875 693 704 823 737 792 49 348 388 873 894 734 204 33 452 85 56 886 6 930 141 58 232
9 580 1000 457 192 755 615 882 652 695 93 255 562 606 93 143 828 97 294 684 35 798 631 72 235 346 574 32 540 8
5 294 363 143 352 444 667 453 873 723 921 866 71 898 865 48 334 20 102 963 475 249 831 440 578 633 148 220 550
0 348 612 186 8 66 995 361 50 851 189 443 192 102 69 425 318 633 135 398 512 520 420 492 318 323 440 189 448 9
694 634 565 262 100 220 310 784 68 68 773 900 940 505 534 21 865 412 801 63 420 706 897 983 100 790 162 408 61
131 247 892 75 790 241 232 190 584 360 987 346 172 123 959
Estos son los números aleatorios impresos y generados.
```

Es hasta aquí en donde podemos apreciar el menú donde están todas las opciones que podemos realizar

Selecciona la opcion que desea realizar:

1. Generar un arreglo de numeros aleatorios
2. Imprimir el arreglo de numeros aleatorios
3. Hacer una busqueda secuencial del arreglo de numeros aleatorios
4. Ordenar el arreglo de numeros aleatorios
5. Ejemplificar la busqueda del arreglo resultando de la opcion 4
6. Calculo del tiempo de ejecucion de las opciones 1 - 5

Una vez aquí podemos imprimir de nuevo los números aleatorios.

Hacemos un ordenamiento de estos números para tener una lista ordenada de ellos.

```
Lista ordenada:
1 2 4 6 7 7 8 8 9 9 10 12 13 13 13 15 16 17 17 18 20 20 21 22 23 24 28 29 29 30 32 32 32 33 33 35 41 47 47 48 48 48 49 4
9 50 52 53 54 56 57 58 58 59 60 62 62 63 65 66 66 68 68 69 70 71 72 72 73 75 78 78 81 82 82 83 85 85 85 87 87 91 93 93 9
3 93 94 95 96 96 97 98 99 100 100 100 100 101 101 101 102 102 104 105 108 108 108 109 113 113 113 114 115 116 120 120 12
2 122 122 123 125 126 127 128 128 129 129 130 131 131 133 134 135 135 137 139 141 141 141 141 141 141 142 143 143 143 144 14
5 145 146 148 148 148 149 154 154 157 159 161 162 162 162 163 166 168 169 170 170 171 172 174 174 174 176 178 179 179 17
9 179 180 181 185 186 187 188 188 189 189 189 190 190 190 191 192 192 192 193 193 194 194 195 197 198 202 202 204 204 20
4 206 206 207 209 211 211 215 215 216 216 218 218 219 220 220 220 220 221 222 222 222 223 224 224 225 227 227 227 228 23
1 231 232 232 232 235 235 236 237 238 238 238 238 239 241 241 241 242 242 243 244 245 245 245 246 247 247 248 249 249 25
1 251 251 253 255 255 257 257 258 260 260 260 261 262 263 265 265 266 267 268 271 272 273 275 276 276 277 278 279 279 28
1 282 283 283 285 286 291 293 294 294 294 294 295 295 295 295 296 299 301 303 303 304 309 309 310 310 310 310 312 312 31
4 314 315 316 317 317 318 318 318 319 320 321 321 321 323 323 323 323 324 328 330 330 330 333 334 335 336 336 337 33
7 337 338 346 346 347 348 348 348 352 352 354 354 354 355 355 357 357 359 359 360 360 360 361 361 361 361 361 363 363 36
3 365 365 367 369 371 371 372 375 375 377 377 378 379 379 380 380 381 381 384 384 386 386 388 389 390 390 395 397 397 39
8 398 398 399 399 399 402 402 403 403 404 407 407 408 408 410 410 411 411 412 412 413 413 413 414 416 416 420 420 420 42
1 423 423 423 424 425 425 430 430 432 433 434 435 436 437 438 439 440 440 441 441 443 444 445 445 446 446 448 448 450 45
1 452 452 453 453 456 456 457 457 459 460 462 462 463 464 467 468 468 472 472 475 475 475 476 476 477 478 480 481 482 48
3 484 484 487 488 488 488 488 489 490 492 494 497 499 499 500 500 503 503 504 505 506 507 509 512 513 513 514 515 51
6 517 520 523 523 527 527 528 533 534 537 538 538 540 540 541 543 544 545 546 547 547 550 553 553 554 555 555 557 55
8 559 560 560 561 562 562 562 564 565 565 565 566 566 567 568 568 571 572 572 573 573 574 574 574 576 576 577 577 578 57
8 578 579 580 580 581 583 584 585 590 593 594 594 595 595 595 596 596 596 596 601 602 602 603 604 605 606 606 606 60
6 608 610 610 610 612 612 612 612 612 613 614 614 615 616 617 621 621 621 621 623 624 624 625 628 631 633 633 633 63
3 634 634 634 634 635 646 647 648 648 648 648 648 650 651 651 652 652 658 659 661 661 661 663 665 666 667 671 672 672 67
2 672 674 674 677 678 679 680 682 684 684 685 686 689 693 693 694 695 695 698 700 700 700 700 700 701 701 702 702 703 70
4 704 704 705 706 710 711 712 712 714 716 717 720 721 722 722 722 723 724 725 725 725 725 726 726 728 729 729 730 730 73
3 734 734 734 735 735 735 736 736 736 737 738 738 738 739 739 742 744 745 745 746 746 747 747 748 748 748 750 750 750 75
1 751 751 751 752 754 755 758 760 761 762 766 768 768 768 769 770 770 771 771 772 773 777 778 779 782 784 784 787 790 79
0 790 792 793 795 797 797 798 798 798 799 800 801 802 806 807 807 808 811 813 813 814 814 818 820 822 823 823 825 82
6 827 827 828 828 828 829 830 831 836 840 842 842 843 847 848 849 850 851 852 854 855 856 856 856 857 858 859 860 861 86
1 862 862 862 862 863 863 863 865 865 866 867 867 867 868 869 870 870 870 872 873 873 873 874 874 875 876 877 878 880 88
```

Teniendo esta lista ordenada podemos ejemplificar una búsqueda con algoritmos de búsqueda secuencial y el cálculo de tiempo que cada función toma para realizar sus operaciones y mostrarlas en pantalla.

Usamos la función main para poder generar inicialmente los números

```

int main() {
    int n;
    printf("Ingrese el tamaño de 1000 números aleatorios ");
    scanf("%d", &n);

    if (n <= 0 || n >= 1001) {
        printf("El tamaño debe ser de 1000.\n");
        return 1;
    }

    srand(time(NULL));

    int arreglo[n];

    for (int i = 0; i < n; i++) {
        arreglo[i] = rand() % 1000 + 1;
    }

    printf("Números antes de ordenar:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arreglo[i]);
    }
    printf("\n");
}

```

A partir de aquí hacemos llamados por medio de funciones switch a cada función

```

int opcion;
printf("Estos son los numeros aleatorios impresos y generados. \n\n");
printf("Selecciona la opcion que desea realizar: \n\n");
printf("1. Generar un arreglo de numeros aleatorios\n");
printf("2. Imprimir el arreglo de numeros aleatorios\n");
printf("3. Hacer una busqueda secuencial del arreglo de numeros aleatorios\n");
printf("4. Ordenar el arreglo de numeros aleatorios\n");
printf("5. Ejemplificar la busqueda del arreglo resultando de la opcion 4\n");
printf("6. Calculo del tiempo de ejecucion de las opciones 1 - 5\n");
scanf("%d", &opcion);

switch(opcion) {
    case 1:
        Generador(arreglo, n);
        break;
    case 2:
        Impresion(arreglo, n);
        break;
    case 3:
        Busqueda(arreglo, n);
        break;
    case 4:
        Ordenamiento(arreglo, n);
        break;
    case 5:
        Ejemplifica(arreglo, n);
        break;
    case 6:
        Calculo(arreglo, n);
        break;
    default:
        printf("Ingrese una opcion valida del 1 al 6, por favor.\n");
        return 1;
}

```

Una vez teniendo eso podemos imprimir la lista con las operaciones aplicadas

```

printf("Lista ordenada:\n");
for (int i = 0; i < n; i++) {
    printf("%d ", arreglo[i]);
}
printf("\n");

```